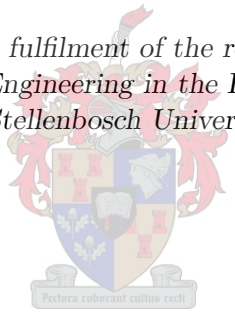
Flight Control System for an Autonomous Parafoil

by

Gideon van der Kolf

Thesis presented in partial fulfilment of the requirements for the degree
Master of Science in Engineering in the Faculty of Engineering
at Stellenbosch University



Supervisor:

Dr I.K. Peddle

Department of Electrical and Electronic Engineering

December 2013

Declaration

By submitting this thesis electronically, I declare that the entirety of the work contained therein is my own, original work, that I am the owner of the copyright thereof, (unless to the extent explicitly otherwise stated) that reproduction and publication thereof by Stellenbosch University will not infringe any third party rights and that I have not previously in its entirety or in part submitted it for obtaining any qualification.

December 2013

Abstract

This thesis presents the development of a flight control system (FCS) for an unmanned, unpowered parafoil and the integration with an existing parafoil system in collaboration with a team at the University of Cape Town (UCT). The main goal of the FCS is to autonomously guide the parafoil from an arbitrary deployment position to a desired landing target. A non-linear 8 degrees of freedom (8-DOF) parafoil model by C. Redelinghuys is incorporated into a MATLAB Simulink simulation environment. The non-linear model is numerically linearised and modal decomposition techniques are used to analyse the natural modes of motion. All modes are determined to be stable but a poorly damped lateral payload relative twist mode is present which causes large payload yaw oscillations. The FCS is divided into stability augmentation, control and guidance subcomponents. Stability augmentation is proposed in the form of a yaw rate damper which provides artificial damping for the oscillatory payload twist mode. For control, a yaw rate controller is designed with the aim of a fast response while not exciting the payload twist oscillation. Subsequently, an existing guidance method is implemented for path following. Autonomous path planning and mission control logic is created, including an energy management (EM) method to eliminate excess height and a terminal guidance (TG) phase. The TG phase is the final turn before landing and is the last chance to influence landing accuracy. A TG algorithm is implemented which generates an optimal final turn and can be replanned en route to compensate for unknown wind and other disturbances. The FCS is implemented on existing avionics, integrated with the parafoil system and verified with hardware in the loop (HIL) simulations. Flight tests are presented but are limited to remote control (RC) tests that verify the integration of the avionics and the parafoil system and test preliminary FCS components.

Uittreksel

Hierdie tesis dra die ontwikkeling voor van ‘n vlug-beheerstelsel (VBS) vir ’n onbemande, onaangedrewe valskerm-sweeftuig (*parafoil*), asook die integrasie daarvan met ’n bestaande stelsel. Die projek is in samewerking met ’n span van die Universiteit van Kaapstad (UCT) uitgevoer. Die VBS se hoof doel is om die sweeftuig outonoom vanaf ’n arbitrêre beginpunt na ’n gewensde landingsteiken te lei. ’n Nie-lineêre 8 grade van vryheid sweeftuig model deur C. Redelinghuys is in die MATLAB Simulink omgewing geïnkorporeer. Die nie-lineêre model is numeries gelineariseer om ’n lineêre model te verkry, waarna die natuurlike gedrag van die tuig geanaliseer is. ’n Swak gedempte laterale draai ossillasie van die loonvrag is geïdentifiseer. Die VBS is opgedeel in stabiliteitstoevoeging, beheer en leiding. ’n Giertempo-demper (*yaw rate damper*) is as stabiliteitstoevoeging om die loonvrag ossillasie kunsmatig te demp, voorgestel. ’n Giertempo-beheerder is ontwerp met die klem op ’n vinnige reaksie terwyl die opwekking van die loonvrag ossillasie terselfdetyd verhoed word. Daarna is ’n bestaande metode vir trajekvolging geïmplementeer. Outonome padbeplanning en oorhoofse vlugplan logika is ontwikkel, insluitend ’n energie-bestuur (EB) metode, om van oortollige hoogte ontslae te raak, asook ’n terminale leiding (TL) metode. Die TL fase verwys na die finale draai voor landing en is die laaste kans om die landingsakkuraatheid te beïnvloed. ’n Bestaande TL algoritme is geïmplementeer wat ’n optimale trajek genereer en in staat is om vir wind en ander versteurings te kompenseer deur die trajek deurgaans te herbeplan. Die VBS is op bestaande avionika geïmplementeer, met die sweeftuigstelsel geïntegreer en met behulp van hardeware in die lus (HIL) simulasies geverifieer. Vlugtoetse is voorgedra, maar is egter beperk tot radio beheer vlugte wat die korrekte integrasie van die avionika en die voertuig toets, asook ’n beperkte aantal voormalige VBS toetse.

Contents

List of Figures

List of Tables

Nomenclature

Abbreviations and Acronyms

8-DOF	8 Degrees Of Freedom
AP	Autopilot
CAN	Controller Area Network
CL	Closed Loop
EM	Energy Management
ESL	Electronic Systems Laboratory
FA	Final Approach
FCS	Flight Control System
GR	Glide Ratio
GS	Ground Station
HC	Heading Controller
HIL	Hardware In the Loop
HILS	Hardware In the Loop Simulation
IMU	Inertial Measurement Unit
L/D	Lift to Drag (ratio)
LSB	Least Significant Bit
MSB	Most Significant Bit
NPF	Non-linear Path Follower (guidance method)
OBC	On-Board Computer
OL	Open Loop
OpTG	Optimal Terminal Guidance
PI	Proportional Integral (control scheme)
RC	Remote Control
RF	Radio Frequency
SD	Secure Digital (memory card format)

SP	Safety Pilot
SU	Stellenbosch University
TG	Terminal Guidance
UCT	University of Cape Town
XT	Cross-track (guidance controller)
YRC	Yaw Rate Controller
YRD	Yaw Rate Damper

Greek Letters

δ_a	Asymmetric brake deflection
δ_s	Symmetric brake deflection
δ_l	Left brake deflection
δ_r	Right brake deflection
γ	Glide path angle
ω_n	Natural frequency in rad/s
τ	Exponential decay time constant
λ	Eigenvalue or pole of a plant
$\mathbf{e_i}$	Eigenvector of the ith pole
ζ	Damping ratio
ψ	Yaw Euler angle
θ	Pitch Euler angle
ϕ	Roll Euler angle

Lowercase Letters

c	Number of EM circles (could be a fraction)
n	Integer number of EM circles
p_p	Parafoil (canopy) body roll rate
p_L	Payload body roll rate
q_p	Parafoil (canopy) body pitch rate
q_L	Payload body pitch rate
r_p	Parafoil (canopy) body yaw rate
r_L	Payload body yaw rate

Uppercase Letters

L_1	Distance to non-linear guidance method reference point
V_h	Horizontal speed
V_v	Vertical speed

Subscripts

A	Parafoil (canopy) aerodynamic axes
I, i	Inertial axes
H	Hinge axes
L	Payload body axes
p	Parafoil (canopy) body axes
w	Parafoil (canopy) wind axes

Syntax and Style

$\mathbf{x}$	The vector $\mathbf{x}$ (usually lowercase)
$\mathbf{A}$	The matrix $\mathbf{A}$ (usually uppercase)

Acknowledgements

I would like to thank the following people for their invaluable contributions and involvement with this project:

- The UCT Team – Prof Redelinghuys, Steven, Jordan, Tracy, Roberto, Prof Attie Grunwald and Gavin. Thank you for making me feel welcome, assisting me and allowing me to be a part of the exciting project.
- Dr Iain Peddle, my study leader, for the guidance, insight and motivation. See you on the road.
- The CSIR, for the financial support and Derrick van Wyk, for your patience, understanding and the roll you played in enabling me to conduct my studies.
- Fellow students at the ESL who made the past few years an enjoyable experience, as well as Mr Japie Engelbrecht, the self-proclaimed parafoil enthusiast, for your enthusiastic interest in the project.
- My parents, for your continued support, love and prayers.
- My awesome girlfriend, Alex Schulze, for your never-ending love, support and motivation.

Chapter 1

Introduction

1.1 Background

The field of unmanned aerial vehicle (UAV) research is very popular today as many practical uses exist for UAVs. The concept of parafoils is not new and autonomous parafoils have been used in a variety of scenarios. The Electronic Systems Laboratory (ESL) at Stellenbosch University (SU) has been conducting UAV research for several years resulting in a well established in-house UAV capability. Previous research projects include fixed wing and rotary wing UAVs which are continually being expanded with new research projects.

A team under guidance of C. Redelinghuys at the Department of Mechanical Engineering, University of Cape Town (UCT) has developed a reconfigurable flight testing system for parafoils with which parafoil flight tests can be conducted in an experimentally controlled environment.

This project will use the UAV expertise in the ESL as a base to develop a flight control system (FCS) and integrate it with the existing vehicle in collaboration with the team at UCT to enable autonomous flight.

1.2 Introduction to Parafoils

A parafoil system, sometimes referred to as a parafoil-payload system or simply a parafoil, is a gliding parachute like vehicle consisting of an air inflated canopy and a payload suspended with lines from the canopy. Figure 1.1 depicts the UCT parafoil in flight. The canopy consists of several cells, open at the leading edge, allowing air to fill and pressurise the canopy, resulting in a large airfoil shaped wing [1]. The configuration of the lines suspending the payload is referred to as the rigging.

1.2.1 Control

The parafoil is typically controlled by pulling down the left or right trailing edge of the canopy with control lines, referred to as deflecting the left or right brake or flap. The left and right brakes are commonly the only control authority on parafoils and can be used either symmetrically or asymmetrically. Asymmetric deflection, denoted by δ_a in this project, refers to deflecting only the left or right brake which enables lateral control of the parafoil. For most parafoils, δ_a causes a rise in drag and a small lift increase on the side of the deflected brake, together with a slight canopy tilt resulting in a turn to the side of the deflection [2]. Slegers and Costello [3] discussed parafoil turn behaviour for different rigging configurations and brake deflections and classified two modes of lateral control: roll and skid steering. Roll steering causes a turn in the opposite direction of deflection while

Figure 1.1 – The UCT parafoil in flight.

Figure 1.2 – Right brake deflection.

skid steering causes a turn in the direction of the brake actuated. Some parafoils exhibit roll steering for small brake deflections and skid steering for larger deflections but a parafoil can be designed or modified to completely eliminate roll steering [3]. This project considers a parafoil with only skid steering. Figure 1.2 depicts the parafoil from the rear with a right brake deflection clearly visible.

Symmetric deflection (δ_s) refers to deflecting both brakes equally and is the longitudinal control of the parafoil. δ_s causes a transient motion directly after deflection followed by a slightly altered steady state glide slope if deflection is held. The transient motion is exploited for the flare manoeuvre in which δ_s is fully deflected right before landing, reducing velocity for a softer ground impact. The steady state effects of δ_s are usually regarded as ineffective, which will also be illustrated in Section 2.4.2, leaving lateral control to be considered as the only control in most cases [2].

Some research has been done on alternative control methods in order to improve the control authority of parafoils. Slegers et al. [2] introduced a technique of varying the canopy incidence angle by changing the longitudinal rigging of the parafoil and payload dynamically in flight in order to obtain better glide slope control. Gavrilovski et al. [4] introduced glide slope control by adding aerodynamic spoilers into the canopy, including a flap spoiler on the lower surface of the canopy and a slit spoiler on the upper canopy surface. While these methods provide useful longitudinal control, they require the addition of actuation and modification to the parafoil setup which is not considered for this project.

Research shows that parafoils typically possess an oscillatory twist motion of the payload relative to the canopy, especially when sudden turning manoeuvres are performed [5], [6]. Slegers [5] investigated the effects of the relative motion between the parafoil canopy and payload on the control of parafoils and showed that persistent payload yaw oscillations relative to the canopy can be eliminated by reducing feedback gains.

1.2.2 Typical Flight Plan

The parafoil considered in this project is unpowered, meaning no engine is present on-board the payload to produce a forward thrust. With no direct means of altitude control, path planning and guidance plays an important role. The typical parafoil mission starts with the parafoil being deployed at an arbitrary height and position away from a desired landing target. A trajectory must be flown which reaches the landing target while taking the initial

height and glide slope of the parafoil into account. A parafoil flight is often divided into the following phases:

- Launch
- Homing
- Energy management (EM)
- Terminal guidance (TG)
- Final approach (FA)
- Flare and landing

Homing refers to flying closer to the landing target and may include user-defined waypoints. However, this may result in excess height which needs to be eliminated before landing is possible. The process of eliminating the excess height is called energy management (EM) and is achieved by flying some manoeuvre. EM and homing may be in any order and is followed by the TG phase, which is the last chance to influence the path of the parafoil, after which the FA is a straight approach leading to the landing target. Some strategies incorporate the TG phase into the EM phase. Prior to landing a flare manoeuvre is performed by applying full δ_s over a very short period. Forward and downward velocities decrease immediately after the deflection, providing a short window of time ideal for landing, after which velocities increase again and a phugoid motion occurs [7]. Flaring into a headwind is advantageous as a further reduction in velocities result, enabling a softer landing.

Several different EM and TG strategies have been used for autonomous parafoils. Yakimenko and Slegers [8] use a "loitering" (energy management) phase defined by 4 waypoints, followed by a TG phase in which a 180° turn, computed with an optimal control algorithm, is performed to the FA. Toohey [9] uses constant circles for EM after which a predefined turn is performed into the FA phase. Jann [10] uses a "T-approach" where a T-shaped pattern, defined by certain waypoints, is flown for EM followed by a final turn point into the FA. The T-formed pattern allows waypoints to be moved to alter the path to compensate for altitude errors. Rademacher [11] uses a Dubins path portion for homing and EM and plans a minimum control-energy trajectory for the TG phase using optimal control theory.

While guidance methods are usually designed to be robust against wind disturbances, the presence of unaccounted for wind will cause the flight time to differ from the expected time. Frequent replanning of the flight path could compensate for the unknown wind, but having a knowledge of the wind velocity is advantageous. One option is having a priori knowledge of the wind velocity. Instrumentation at the ground station can continually feed wind measurements to the control system during flight via the RF link. Alternatively, a measurement can be given to the control system just before the launch and assumed to be constant throughout the flight. If it is assumed that the parafoil moves at a constant airspeed with zero sideslip angle, the known airspeed can be used in combination with the ground speed measurement from the GPS to determine the wind vector. The airspeed can be measured on-board using a pitot tube, if available. Rademacher [11] uses simple airspeed measurements to estimate wind on-board. Roos [12] measures the average wind vector by flying a constant turn rate and measuring the displacement of the start and end point after a full turn has been completed. This method can be incorporated into the energy management phase with little effort. Ward et al. [13] and Jann [10] estimate the wind vector from only GPS data. Jann [10] uses a non-linear estimation filter to estimate the wind components. However, the estimations might take a long time to converge to the actual values and the speed and quality of the estimation depends on the flight pattern. Ward et al. [13] breaks the flight into constant control deflection segments from which wind is estimated. The

estimation also depends on the flight path, where a straight path will not contain good wind estimates.

1.2.3 Uses of Parafoils

Parafoils are used for airdrop missions where supplies and equipment need to be delivered to remote locations which may be dangerous or hard to reach by land. The gliding and steering capability of parafoils offer several advantages over conventional parachutes [11]. In addition, parafoils are typically lightweight and compact before deployment and can carry reasonable payloads, impacting the ground at a low speed. Other uses include recovery of flying vehicles.

Regarding autonomous parafoil projects, there are several documented and ongoing projects in literature. Toohey [9] developed a small parafoil vehicle for precision delivery. Draper Laboratories [14] conducts the Joint Precision Airdrop System (JPADS) program with the goal to develop an autonomous guidance, navigation and control (GNC) system enabling precision airdrop capability with a wide payload capacity range. Slegers and Yakimenko performed research in several areas of autonomous parafoil capability including optimal control for the terminal guidance phase [8] and developing and testing a miniature aerial delivery system [15]. Rademacher [11] conducted extensive research on in-flight trajectory planning and guidance for a wide range of parafoils. Thomas Jann at the Institute of Flight Systems of the German Aerospace Center (DLR) [10] developed a GNC system with a "T-Approach" guidance algorithm.

1.3 Available Resources

The following resources are available for this project:

- ESL avionics
- ESL ground station hardware and software
- Hardware in the loop (HIL) simulation framework
- Non-linear 8 degrees of freedom (8-DOF) mathematical model and MATLAB simulation by Redelinghuys [16]
- The UCT reconfigurable flight testing system

The ESL avionics, ground station and HIL simulation framework is a collection of well tested hardware and software components that were developed and used in previous UAV projects at the ESL over several years. However, these resources are only configured for fixed wing and rotary wing UAVs and will have to be configured for the parafoil.

Several mathematical parafoil models have been introduced in the literature which differ in degrees of freedom and complexity. The model used in this project is an 8-DOF model which takes the relative motion between the parafoil canopy and payload into account. This specific model will be used since this project is collaborating with the author, from which a computer simulation based on the model was obtained.

The reconfigurable flight testing system of UCT includes

- the parafoil,
- the payload box with steering unit,
- UCT ground control station for manual remote control (RC), and

- a parafoil launching system.

These components were designed by the UCT team and is part of an on-going project to establish a reliable parafoil testing environment and a RC capable parafoil [17]. Figure 1.3 depicts the payload box which houses a steering unit and the ESL avionics. The steering unit contains servo motors and a controller to wind up the brake lines for brake deflection.

Figure 1.3 – The payload box.

Figure 1.4 – The UCT parafoil launcher system.

The parafoil and payload mounted on the launcher system is shown in Figure 1.4. The launcher has a catapult like arm which swings upward, inflating the canopy after which the payload box is accelerated forward to produce a coordinated launch of the entire system. A launch sequence is shown in Figure 1.5.

Figure 1.5 – The launch sequence.

1.4 Goals, Scope and Project Outline

The main objective of the project is to develop and implement a flight control system (FCS) on the available hardware and integrate it with the existing vehicle in order to enable autonomous flight. The FCS must be able to autonomously plan an appropriate path and guide the parafoil to land at a desired target. The following need to be considered:

- The parafoil will be deployed at an arbitrary height and distance away from the desired landing target. Homing, EM and TG phases need to be implemented.
- Despite unknown disturbances and path following errors, the landing target must be reached as accurately as possible.
- The natural modes of motion of the parafoil must be analysed to determine whether stability augmentation is needed.

- The FCS is to be implemented on the existing ESL avionics and integrated with the UCT parafoil system.
- The flare manoeuvre and wind estimation are not considered part of the scope of this project. An accurate flare manoeuvre is sensitive to timing and requires additional equipment to accurately sense the vehicle height.

Chapter 2 starts the process by discussing the parafoil model and using the computer simulation to illustrate some relevant parafoil characteristics. Chapter 3 will continue by deriving a linear model and using it to analyse the natural modes of motion of the parafoil. In Chapter 4, the inner FCS controllers and guidance method will be designed, after which path planning and mission control will be addressed in Chapter 5. After the FCS design, the hardware and software used and created in this project is discussed in Chapter 6. Subsequently, the FCS will be tested in HIL simulations to verify the implementation on the hardware. Flight tests will be presented in Chapter 8, followed by conclusions and recommendations.

Chapter 2

Mathematical Model

For this project, an 8 degrees of freedom (8-DOF) mathematical parafoil model, developed by Redelinghuys at the University of Cape Town (UCT), is used [16]. The model is implemented as a computer simulation in the MATLAB/Simulink environment. The simulation is essential to the design and testing of the flight control system. Since the parafoil vehicle is being developed in parallel to this project and not available for flight test purposes in the initial phase of the project, the simulation serves as a source of insight and a convenient experimentation and testing medium.

In the following sections, an overview of the parafoil model and computer simulation is given, after which the simulation is used to illustrate some of the important parafoil characteristics in Section 2.4.

2.1 Axis Systems

The mathematical model employs several right-handed axis systems which are summarised in Table 2.1.

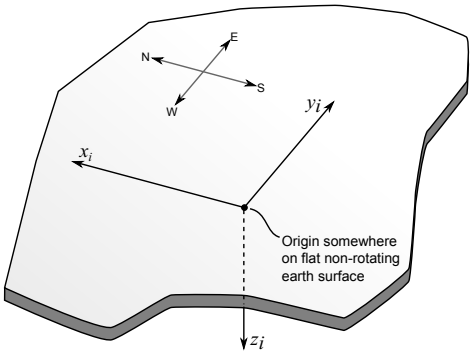


Figure 2.1 – Inertial axis system.

First, consider the approximate inertial axis system for the purposes of this project, depicted in Figure 2.1. This is a right-handed axis system with its origin chosen at some convenient location on the earth surface. The earth is assumed to be flat and non-rotating, which is reasonable since flight ranges are relatively small and the typical angular rotations of the vehicle are much greater than that of the earth. Redelinghuys [16] chooses the inertial axes origin at sea level since the parafoil model takes air density into account. For control system

Axis system	Description
Inertial axes	Origin at sea level x_i pointing north y_i pointing east z_i pointing down
Payload body axes	Origin at payload mass centre x_L positive forward y_L positive in starboard direction z_L positive down
Hinge axes	Origin referred to "hinge", midway between suspension line coupling points x_H parallel to x_L y_H parallel to y_L z_H parallel to z_L
Parafoil body axes	Origin at parafoil canopy mass centre x_p completes right-handed system y_p positive in starboard direction z_p positive downward, pointing to hinge
Parafoil aerodynamic axes	Origin at parafoil nose in symmetry plane x_A points forward, parallel to wing chord y_A positive in starboard direction, parallel to y_p z_A completes right-handed system
Parafoil wind axes	Origin at parafoil nose in symmetry plane x_w pointing into relative wind y_w to the right in $x_w - y_A$ plane z_w completes right-handed system

Table 2.1 – Axis systems used [16].

purposes, it is convenient to choose the origin to coincide with the launch point or landing target. The positive z_i axis points downward to the centre of the earth. The x_i axis is perpendicular to the z_i axis and points positive to north, while the y_i axis is positive to east.

The inertial axis system serves as a fixed reference frame relative to which the position and rotation of the other axis systems are described. The axis systems related to the parafoil-payload system are the payload body axes (referred to as UAV body axes by [16]), hinge axes, parafoil (canopy) body axes, parafoil aerodynamic axes and the parafoil wind axes, illustrated in Figure 2.2 and summarised in Table 2.1.

The payload body axes, denoted with subscript L, originate at the payload mass centre, with the z_L axis positive downward, x_L positive forward and y_L positive to the starboard (right) side. The payload position is described by three coordinates X_L, Y_L and Z_L which describe the north, east and down positions, respectively, of the payload body axes origin relative to the inertial axes. Three Euler angles describe the payload orientation: pitch, θ, a rotation about the y_L axis; roll, ϕ, about the x_L axis; and yaw, ψ, about the z_L axis. The positive rotation directions are defined according to the right-hand convention. The payload orientation relative to inertial axes is described by the rotation of the payload body axes with respect to the inertial axes in the Euler 3-2-1 sequence (ψ, θ, ϕ sequence) as described in [18, p. 10].

The hinge axes, x_H, y_H and z_H, are parallel to the payload body axes x_L, y_L and z_L, respectively, with its origin, referred to as the "hinge", midway between the suspension line coupling points as indicated in Figure 2.2.

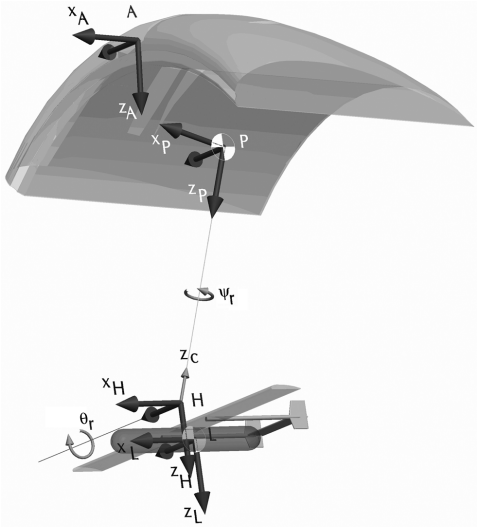


Figure 2.2 – Axis systems of the parafoil-payload system [16].

The parafoil body axes originate at the canopy mass centre, with the z_p axis pointing down toward the hinge point, y_p down the starboard canopy wing and x_p completing the right-handed axis system. The orientation of the canopy relative to the payload is defined by a rotation θ_r about the y_H axis, followed by a rotation ψ_r about the z_p axis required to coincide the hinge axes with the parafoil body axes [16], with positive directions indicated in Figure 2.2. The relative roll between the canopy and payload is assumed to be zero.

The parafoil aerodynamic axes origin is at the nose of the parafoil canopy, laterally in the centre (in the plane of symmetry). The x_A axis is positive forward, parallel to the chord of the wing. y_A is positive right, parallel to the y_p axis, and z_A points downward, perpendicular to the wing chord, completing the right-handed system. The angle between the parafoil body axes and the aerodynamic axes, θ_{Ap}, is referred to as the angle of incidence or the canopy rigging angle. The angle is determined by the rigging, i.e. the combination of lengths of the suspension lines and is chosen based on the desired glide characteristics. θ_{Ap} is displayed in Figure 2.3.

Lastly, the parafoil wind axes originate at the canopy nose in the plane of symmetry, similar to the aerodynamic axes. The x_w axis points into the relative wind, y_w points to the right in the x_w-y_A plane and z_w completes the right-handed system.

All axis systems are orientated similar to the payload body axes mentioned above, using the Euler 3-2-1 rotation. Transformation matrices associated with the axis systems are presented in [16, Appendix A]. While [16] employs all of the axis systems in the derivation and computation of the parafoil model, for control and analysis purposes, this project is mostly concerned with the payload position and orientation described by the payload body axes and the parafoil relative orientation, described by the parafoil body axes.

2.2 Model Overview

The 8-DOF parafoil model by Redelinghuys [16] describes the motion of a parafoil canopy suspending a payload, both of which are assumed rigid, with suspension lines that do not stretch. 6 of the 8 degrees of freedom represent the position and orientation of the payload body axes (originating at the payload mass centre) with regards to inertial axes, while the remaining 2 describe the yaw and pitch rotation of the canopy relative to the payload. Relative roll is assumed to be zero. The payload position is given by coordinates X_L, Y_L

and Z_L and the attitude by ψ_L, θ_L and ϕ_L, as described in the previous section. The two additional states are the relative pitch θ_r and yaw ψ_r of the parafoil canopy with regards to the payload body.

Redelinghuys [16] uses a quasi-Hamiltonian formulation to develop the equations of motion. This avoids tedious differentiations of generalised momenta with respect to time normally found in a Lagrangian approach [16]. The state vector resulting from the formulation is given by

$$\mathbf{x} = [\ p_x\ p_y\ p_z\ p_\psi\ p_\theta\ p_\phi\ p_{\theta r}\ p_{\psi r}\ p_{zc}\ x\ y\ z\ \psi\ \theta\ \phi\ \theta_r\ \psi_r\ z_c\ ]^T \tag{2.2.1}$$

where the first 9 components are the generalised momenta and the last 9 are generalised coordinates. p_x, p_y and p_z represent the linear momentum of the combined parafoil-payload system in the respective directions indicated by the subscripts. p_ψ, p_θ and p_ϕ represent the angular momenta of the system with respect to a point coinciding with the payload mass centre. $p_{\theta r}$ and $p_{\psi r}$ are canopy angular momenta with respect to a point coinciding with the hinge point. x, y, z, ψ, θ and ϕ represent the payload position and orientation in inertial axes, while θ_r and ψ_r represent the relative rotation of the canopy with respect to the payload. p_{zc} represents the linear momentum of the canopy directed from the hinge to the canopy mass centre [16]. p_{zc} and z_c are used to determine the internal line force between the payload and canopy. z_c represents the displacement of the hinge and leads to an equation of constraint, $z_c = 0$.

The model includes the influence of canopy warping on aerodynamic loads when brakes are applied and accommodates for dynamic pressure. Redelinghuys [16] gives a structure for calculating and including aerodynamic parameters from data generated by a commercially available computational fluid dynamics code.

Symmetric and asymmetric brake deflection and wind input may be chosen and canopy and payload dimensions and aerodynamic properties can be specified. In addition to the states in Equation (2.2.1), the model enables calculation of angle of attack, sideslip angle and angular rates.

For more in-depth information and the derivation of the model, [16] should be consulted. For the purposes of this project, focus is now shifted to the computer simulation implementation.

2.3 Computer Simulation

The 8-DOF parafoil model was originally implemented as a MATLAB simulation by the model author. The simulation was later ported to a C++ implementation by A. Grunwald. Eventually, for this project, the simulation was ported to a Simulink C s-function, enabling it to be used as an add-in block in a Simulink simulation. The Simulink s-function executes quickly compared to the MATLAB simulation, allowing lengthy simulations, at a high sample rate, to be carried out in a matter of seconds. In addition, countless simulations performed with the original MATLAB simulation led to the creation of a simple graphical user interface (GUI) front-end to the simulation which allows a fast, user friendly way of initialising and running a simulation without the need to manually edit various script files to set the simulation parameters.

2.4 Parafoil Characteristics

Table 2.2 describes the physical characteristics of the parafoil system for which the simulation is configured. A more detailed listing of the associated parameters and aerodynamic coefficients is presented in Appendix A. The Simulink simulation is used to illustrate some typical flight characteristics of the parafoil in the following sections.

Property	Value
Canopy wing chord	1.56 m
Projected wing span	3.9 m
Canopy mass (including lines)	1.5 kg
Payload mass	25.6 kg

Table 2.2 – Parafoil and payload physical properties.

2.4.1 Steady State, Homogeneous Flight

Firstly, steady state homogeneous flight (i.e. control input and wind disturbances are zero) of the parafoil is observed. The trim flight condition is calculated by the simulation based on the geometric, aerodynamic and rigging properties of the parafoil and is summarised in Table 2.3. The conditions are depicted graphically in Figure 2.3. The trim condition is dependent on the parafoil rigging which refers to the suspension lines lengths and the canopy incidence angle and is discussed in detail by [7].

Property	Symbol	Trim value
Total glide velocity	V	13.94 m/s
Vertical velocity	V_v	5.14 m/s
Horizontal velocity	V_h	12.99 m/s
Glide slope angle	γ	21.63°
Glide ratio	GR	2.52
Canopy angle of attack	α_A	3.53°
Aerodynamic pitch angle (wing chord angle)	θ_A	-18.1°
Canopy body pitch angle	θ_p	-1.79°
Canopy incidence angle (Rotation from A to p axes)	θ_{Ap}	16.3°
Payload pitch	θ_L	-6.11°
Payload angle of attack	α_L	15.52°

Table 2.3 – Parafoil simulation initial trim settings.

2.4.2 Symmetric Brake Deflection

When symmetric brakes, δ_s, are applied, the parafoil experiences a transient motion after which it settles into a new steady state glide slope and glide velocity. Exploiting the transient response is one of the most important uses of δ_s and is referred to as the flare manoeuvre.

The flare manoeuvre is used to reduce velocity for landing, performed by applying a full δ_s step which increases the lift and drag, reducing the horizontal and vertical velocity and thus minimising landing impact [7]. Figure 2.4 shows the simulation results for a flare manoeuvre executed at 5 s, by deflecting δ_s from 0% to 100% over 1 second. Following the δ_s step, the forward and down velocities fall and the parafoil pitches upward. The minimum down velocity of approximately 2 m/s is reached at 7 s which marks the ideal touchdown instant, after which it rapidly increases again. The ideal landing conditions are reached 2 seconds after the flare initiation, in which time the vehicle travelled 6.9 m vertically. The time window for performing an effective landing flare is very small and precise, as seen in the velocity history in Figure 2.4(b). Flaring into a headwind is advantageous since forward velocity is reduced further [7]. After the flare manoeuvre, the parafoil performs a damped phugoid-like motion, characterised by the sinusoidal response in glide path angle, pitch and vertical velocity while the angle of attack remains approximately constant. As the motion dies away the parafoil settles into a new steady state glide slope.

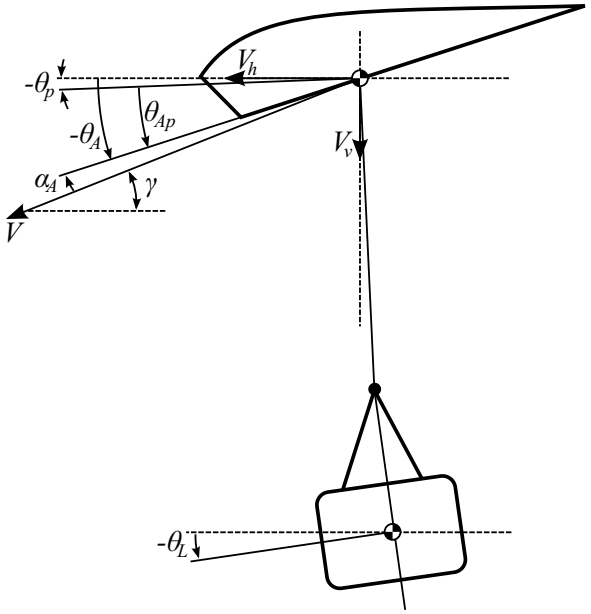
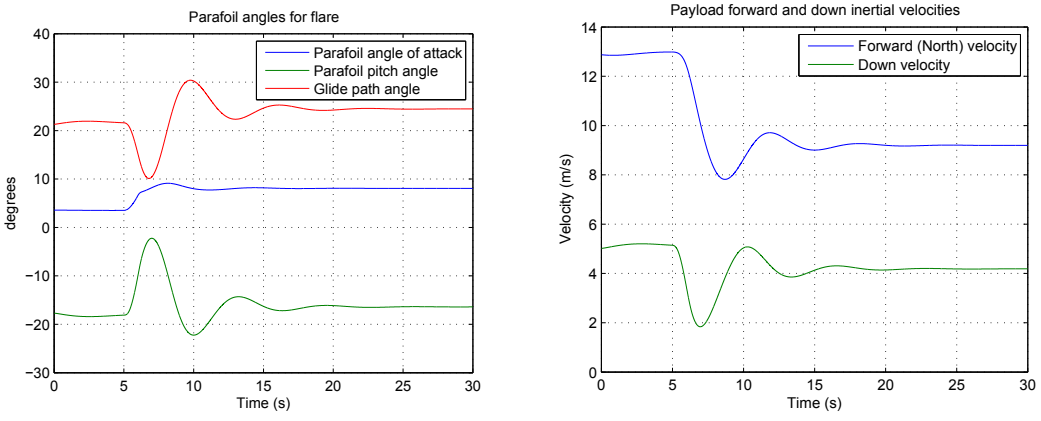


Figure 2.3 – Trim condition.

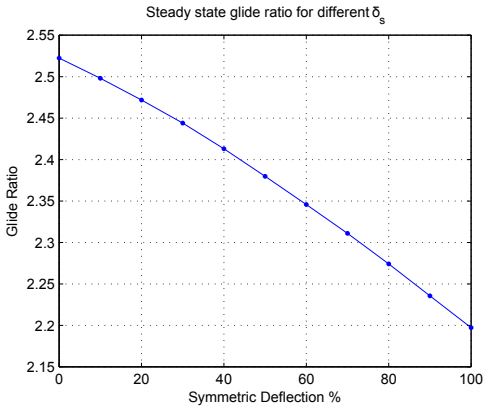


(a) Parafoil angles

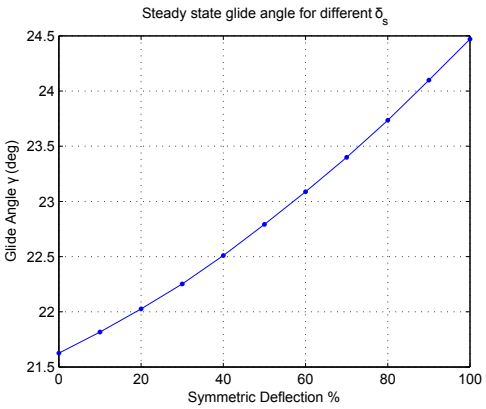
(b) Inertial velocities

Figure 2.4 – Response of flare manoeuvre, 0% to 100% δ_s over 1 second.

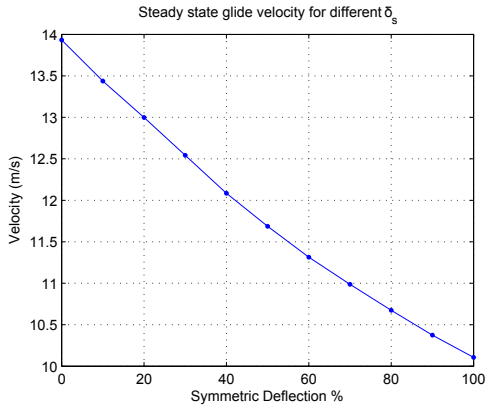
The effect of δ_s on steady state glide slope and glide speed differs between parafoils and is dependent on rigging properties, however the general trend is that steady state glide slope is affected relatively little [2]. δ_s causes very small changes in trim angle of attack and an increase in both lift and drag, resulting in the lift to drag (L/D) ratio remaining nearly constant while glide velocity is reduced [7]. Figure 2.5 illustrates the effect of δ_s on the steady state glide ratio and glide velocity for the current parafoil model in zero wind conditions. Zero to full brake deflection causes a reduction in glide velocity of 3.83 m/s and



(a) Glide ratio



(b) Glide angle γ



(c) Glide velocity V

Figure 2.5 – Parafoil glide characteristics for variation in δ_s.

the glide ratio to fall by 0.32. The glide slope angle, γ, is increased from 21.6° to 24.5°.

The effect of δ_s on the L/D ratio can be increased with rigging changes such as reducing the canopy incidence angle θ_{Ap} [11] or increasing the suspension line length [7]. Lingard [7] also notes that δ_s can have a greater effect on the glide slope in the presence of a headwind. Forward velocity decreases as δ_s increases resulting in a greater effect from the headwind. However, the change in glide slope is still small for the current model, as summarised in Table 2.4 for cases with different wind strengths. W represents headwind strength and $\Delta\gamma$, ΔV and ΔGR indicates change in glide slope angle, change in glide velocity and change in glide ratio, respectively, for a δ_s variation of 0% to 100%.

Other methods of glide slope control are possible, such as shifting the position of the mass centre, as employed by human skydivers [2]. Slegers et al. [2] introduced a technique of

W	$\Delta\gamma$	ΔV	ΔGR
0 m/s	2.84°	-3.83 m/s	-0.32
3 m/s	6.75°	-3.72 m/s	-0.46
5 m/s	12.06°	-3.54 m/s	-0.55

Table 2.4 – Effect of δ_s=0% to 100% on glide slope with a headwind.

varying the canopy incidence angle by changing the longitudinal rigging of the parafoil and payload dynamically in flight in order to obtain better glide slope control. Gavrilovski et al. [4] introduces glide slope control by adding aerodynamic spoilers into the canopy, including a flap spoiler on the lower surface of the canopy and a slit spoiler on the upper canopy surface. However, all of these glide slope control techniques require additional actuators and modification to the parafoil setup.

2.4.3 Asymmetric Brake Deflection

Asymmetric brake deflection, δ_a, is generally the main control authority in autonomous parafoils. The model exhibits what is referred to by [3] as skid steering, where the parafoil will turn in the direction which brake is deflected. When applied, a slight increase of lift and large increase of drag is generated by the deflected brake which induces a yawing moment [11]. The differential drag leads to a sideslip angle, causing a side-force which rolls the canopy. The yaw moment is countered by adverse yaw moment due to yaw rate, finally achieving a steady state turn [11].

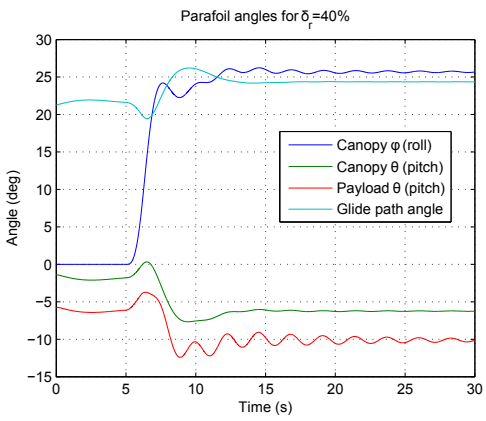
As for δ_s, a δ_a deflection causes a transient motion after which the parafoil settles into a steady state. Figure 2.6 shows simulation results for a 40% right brake (δ_r) step input. Pitch angle, glide angle and velocities, in Figures 2.6(a) and 2.6(e), display similar behaviour to the δ_s response. What is of more concern is the very poorly damped payload yaw rate oscillation observed in Figure 2.6(b). When δ_r is deflected, the canopy settles into a steady yaw rate while a yaw oscillation persists in the payload. For a parafoil suspending a fixed wing unmanned aerial vehicle (UAV) payload, as presented in [16], simulation results[1] for a similar δ_a step input show a drastically more damped payload yaw oscillation which dies away after approximately 2 periods. The increased damping is attributed to the winged UAV payload which provides better lateral aerodynamic damping. For the current system, the lack of payload yaw damping will have to be addressed in the control system design process.

Figure 2.7 combines the steady state results of several simulations with different δ_a. Note that where the sign of the response depends on which brake is deflected, right brake deflection is used, indicated by δ_r. The glide ratio and angle γ is greatly influenced by δ_a, as opposed to the case for δ_s, especially for deflections larger than 50%, as seen in Figures 2.7(a) and 2.7(b). Similar behaviour is indicated for vertical (down) velocity, V_v, in Figure 2.7(c), which increases more drastically as δ_a increases above 50%. δ_a results in a steady state turn (yaw) rate which increases approximately linearly related to δ_a for deflections up to 50%, as illustrated in Figure 2.7(d). Maintaining δ_a below 50% will thus reduce the change in sink rate when turning and ensure a more linear relationship between δ_a and turn rate.

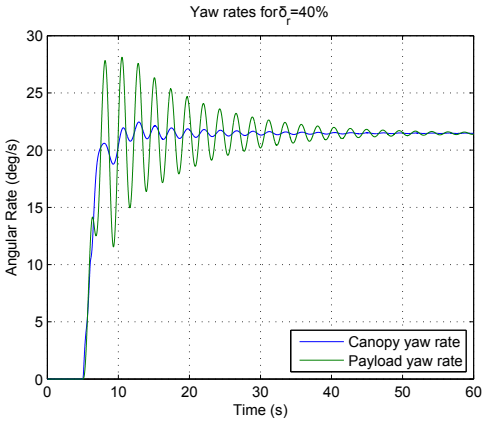
2.5 Summary

This chapter presented the various axis systems employed by the parafoil model, gave an overview of the model and computer simulation and used it to illustrate some typical flight

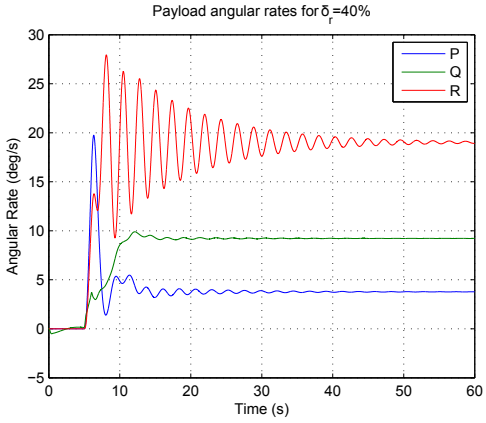
[1]The simulation was performed with the MATLAB model of Redelinghuys, representing a parafoil suspending a UAV as described in [16].



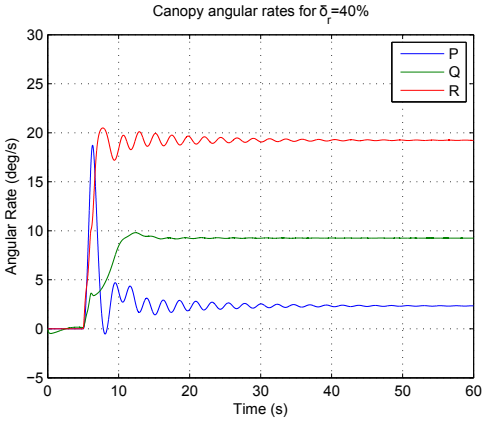
(a) Parafoil angles



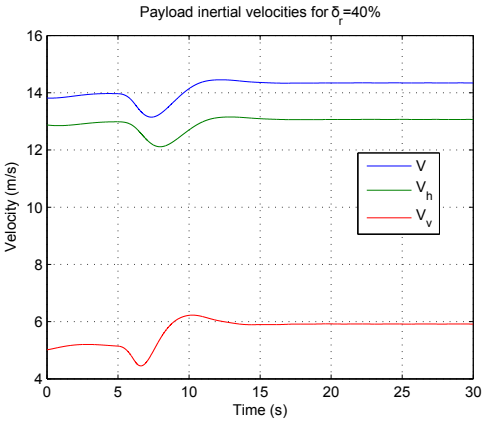
(b) Inertial yaw rates



(c) Payload body angular rates

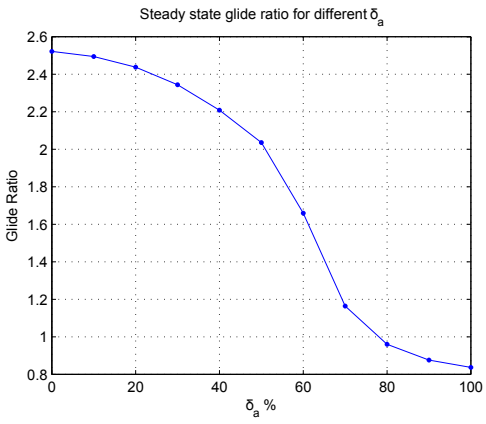


(d) Canopy body angular rates

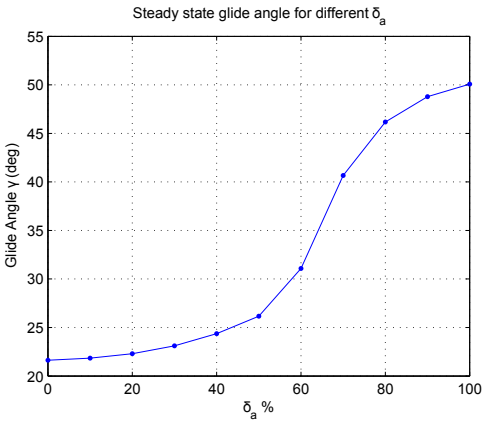


(e) Velocities

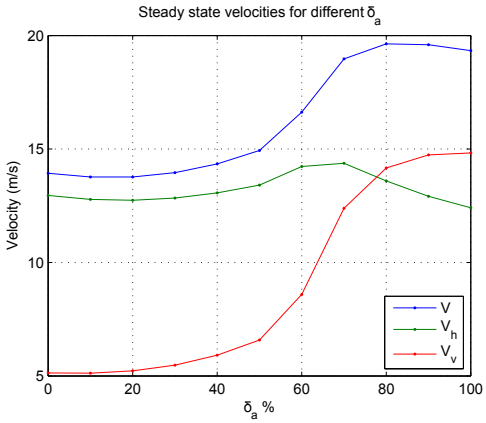
Figure 2.6 – Step response for 40% right δ_a.



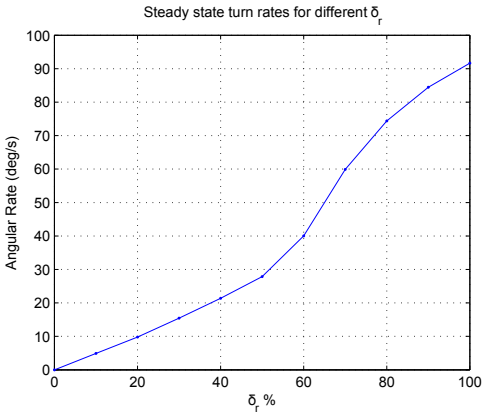
(a) Glide ratio



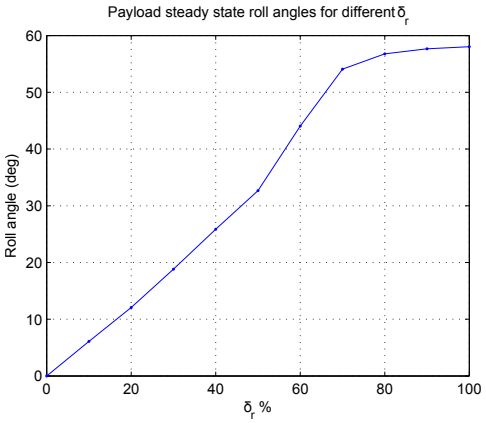
(b) Glide angle γ



(c) Velocities



(d) Turn rate $\dot{\psi}_L$



(e) Roll angle ϕ_L

Figure 2.7 – Steady state characteristics for variation of δ_a.

characteristics of the parafoil. The trim flight conditions were listed in Table 2.3. The transient and steady state responses for both symmetric and asymmetric brake deflection was described and it was shown that, apart from the flare manoeuvre, little useful control is available from δ_s while the main control authority is δ_a. Notable consequences of δ_a, however, are increased sink rate, especially for large brake deflections, and a very poorly damped payload yawing oscillation, induced by a δ_a step.

Chapter 3

Linearising and Analysis

The non-linear 8-DOF parafoil model is an essential tool for testing and evaluating control systems designed for the parafoil. However, for the purpose of analysing the system dynamics and control system design, a linear model is desired. Control system design for linear systems is less complex and valuable insight is gained from analysing the linear model. Although the linearised model resembles the actual parafoil less accurately than the non-linear model, robustness of the control system will compensate for the model inaccuracies.

This chapter presents the method for obtaining the linear model and verifies it by comparing its output to that of the non-linear model. The poles of the linear model are then analysed, which describe the natural modes of motion of the parafoil.

3.1 Linearisation

The 8-DOF parafoil model is a complex and complicated model derived analytically with Lagrangian mechanics. The model has also been implemented reliably as a MATLAB simulation. Linearisation can be done analytically by studying the mathematical model, substituting approximations and simplifying it. Alternatively, a numerical linearisation method can be performed where the linear model is deduced from the MATLAB simulation. The latter method is chosen since it eliminates effort and time required by the former.

3.1.1 Small Disturbance Theory

The desired linear model is described by a set of linear equations [19]:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}, \mathbf{w}) \tag{3.1.1}$$

$$\mathbf{y} = \mathbf{h}(\mathbf{x}) \tag{3.1.2}$$

where $\mathbf{x}$ is a column vector of the state variables, $\mathbf{u}$ is a vector representing the inputs to the system and $\mathbf{w}$ represents the wind disturbance inputs. Note that $\mathbf{f}$ and $\mathbf{h}$ are not functions of time as it is assumed that they do not change significantly over time.

An equilibrium point with reference values $\mathbf{x}_0$, $\mathbf{u}_0$ and $\mathbf{w}_0$ is chosen about which to linearise and

$$\mathbf{f}(\mathbf{x}_0, \mathbf{u}_0, \mathbf{w}_0) = 0 \ . \tag{3.1.3}$$

$\mathbf{x}$, $\mathbf{u}$ and $\mathbf{w}$ are assumed to always be close to the equilibrium point reference values and can be written as the sum of the equilibrium values and small disturbances.

$$\mathbf{x} = \mathbf{x}_0 + \delta\mathbf{x} \tag{3.1.4}$$

$$\mathbf{u} = \mathbf{u}_0 + \delta\mathbf{u} \tag{3.1.5}$$

$$\mathbf{w} = \mathbf{w}_0 + \delta\mathbf{w} \tag{3.1.6}$$

These equations with the small disturbances are substituted into Equation (3.1.1). By expanding the equation and ignoring the higher order terms of the small disturbance components, the equation finally results in [19]:

$$\delta\dot{\mathbf{x}} = \mathbf{f}_{,\mathbf{x}}(\mathbf{x}_0, \mathbf{u}_0, \mathbf{w}_0)\delta\mathbf{x} \ + \ \mathbf{f}_{,\mathbf{u}}(\mathbf{x}_0, \mathbf{u}_0, \mathbf{w}_0)\delta\mathbf{u} \ + \ \mathbf{f}_{,\mathbf{w}}(\mathbf{x}_0, \mathbf{u}_0, \mathbf{w}_0)\delta\mathbf{w} \tag{3.1.7}$$

$$\delta\mathbf{y} = \mathbf{h}_{,\mathbf{x}}\delta\mathbf{x} \tag{3.1.8}$$

with the notation of the functions (called the Jacobian) defined as

$$\mathbf{f}_{,x} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \cdots & \frac{\partial f_n}{\partial x_n} \end{bmatrix} . \tag{3.1.9}$$

For convenience, the Jacobian matrices are simply written as

$$\begin{aligned} \mathbf{F} &= \mathbf{f}_{,\mathbf{x}}(\mathbf{x}_0, \mathbf{u}_0, \mathbf{w}_0) \\ \mathbf{G} &= \mathbf{f}_{,\mathbf{u}}(\mathbf{x}_0, \mathbf{u}_0, \mathbf{w}_0) \\ \mathbf{W} &= \mathbf{f}_{,\mathbf{w}}(\mathbf{x}_0, \mathbf{u}_0, \mathbf{w}_0) \\ \mathbf{H} &= \mathbf{h}_{,\mathbf{x}}(\mathbf{x}_0) \end{aligned}$$

and δ is excluded from the equations for it to be simply written as [19]

$$\dot{\mathbf{x}} = \mathbf{F}\mathbf{x} + \mathbf{G}\mathbf{u} + \mathbf{W}\mathbf{w} \ , \qquad \mathbf{y} = \mathbf{H}\mathbf{x} \ . \tag{3.1.10}$$

Equation (3.1.10) is the state space representation of the linear model. A method of numerical linearisation can now be performed where $\mathbf{F}$, $\mathbf{G}$ and $\mathbf{W}$ are calculated by applying small disturbances from the chosen trim (equilibrium) state of the non-linear model. $\mathbf{H}$ is not calculated in the initial linearisation process as it relates the states to the output and can be chosen arbitrarily in order to observe a desired output of the system.

3.1.2 Numerical Linearisation Method

The following outlines the steps taken to linearise the non-linear parafoil model numerically.

1. A trim flight condition is chosen about which the model will be linearised. The chosen trim condition is straight flight at a constant forward and downward velocity, with no control or wind inputs.

2. The non-linear parafoil simulation is initialised with the state vector for the chosen trim condition, $\mathbf{x}_0$. Control and wind input are set to zero. The simulation is run for one sample instance and the resulting derivative state vector, $\dot{\mathbf{x}}_0$, is recorded.

3. Now, a small perturbation is added to the first element of the initial trim state vector:

$$\mathbf{x}_0 + \delta\mathbf{x}_1 = \begin{bmatrix} x_{0_1} + \delta x_1 \\ x_{0_2} \\ \vdots \\ x_{0_n} \end{bmatrix} \tag{3.1.11}$$

4. The non-linear simulation is initialised with the vector in Equation (3.1.11) as the initial state vector. Control and wind input are set to zero and the simulation is run again for one sample instance. The resulting derivative state vector,

$$\dot{\mathbf{x}}_1 = \dot{\mathbf{x}}_0 + \delta\dot{\mathbf{x}}_1 \tag{3.1.12}$$

is recorded. $\dot{\mathbf{x}}_0$ is the trim derivative state vector and $\delta\dot{\mathbf{x}}_1$ is the small disturbance resulting from $\delta\mathbf{x}_1$.

5. Consequently, from Equation (3.1.7) we have

$$\delta\dot{\mathbf{x}}_1 = \mathbf{f}_{,x}(\mathbf{x}_0, \mathbf{u}_0, \mathbf{w}_0)\delta\mathbf{x}_1 \ . \tag{3.1.13}$$

Recall that the control and wind input were made zero and thus do not appear in the equation. Equation (3.1.13) reduces to

$$\delta\dot{\mathbf{x}}_1 = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} \\ \\ \frac{\partial f_2}{\partial x_1} \\ \\ \vdots \\ \\ \frac{\partial f_n}{\partial x_1} \end{bmatrix} \delta x_1 \tag{3.1.14}$$

from which the first column of $\mathbf{F}$ can be solved.

6. The process is repeated from step 3 for a perturbation in each state, each time calculating the next column of $\mathbf{F}$.

7. Once the entire $\mathbf{F}$ matrix has been computed, the $\mathbf{G}$ and $\mathbf{W}$ matrices are calculated separately in a similar fashion, this time applying control ($\delta\mathbf{u}$) and wind ($\delta\mathbf{w}$) perturbations, respectively.

Sensible values for the perturbations $\delta\mathbf{x}$, $\delta\mathbf{u}$ and $\delta\mathbf{w}$ must be selected. These values should be small enough so that the resulting linear model will resemble the non-linear model with sufficient accuracy, but large enough to prevent inaccuracies due to round-off errors. Changing the magnitude of the perturbations will change the entries and eigenvalues of $\mathbf{A}$ [20]. The effect on the eigenvalues when linearising with different perturbation values is analysed and a value of

$$\delta x = \delta u = \delta w = 1{\times}10^{-7} \tag{3.1.15}$$

was selected to linearise with.

3.1.3 Numerical Linearisation Results

The method described in the previous section was used to perform numerical linearisation of the non-linear parafoil model. The state vector is adopted from the non-linear model [16] as

$$\mathbf{x} = [\ p_x\ p_y\ p_z\ p_\psi\ p_\theta\ p_\phi\ p_{\theta r}\ p_{\psi r}\ p_{zc}\ x\ y\ z\ \psi\ \theta\ \phi\ \theta_r\ \psi_r\ z_c\ ]^T \ . \tag{3.1.16}$$

The states in Equation (3.1.16) were discussed in Section 2.2. The control input vector is given by

$$\mathbf{u} = \begin{bmatrix} \delta_l \\ \delta_r \end{bmatrix} \tag{3.1.17}$$

where δ_l denotes left brake deflection and δ_r denotes right brake deflection, between 0% and 100% inclusive. Symmetric deflection δ_s is achieved by deflecting both δ_l and δ_r equally, while asymmetric deflection δ_a translates to only deflecting one of the two. The convention used is that a positive δ_a relates to a δ_r deflection, while a negative δ_a relates to δ_l.

The wind input vector contains the x (positive north), y (positive east) and z (positive down) wind components in inertial axes,

$$\mathbf{w} = \begin{bmatrix} w_x \\ w_y \\ w_z \end{bmatrix} \ . \tag{3.1.18}$$

The trim condition about which the model is linearised is described in Table 2.3. The trim state vector is

$$\mathbf{x}_0 = \begin{bmatrix} p_{x0} \\ p_{y0} \\ p_{z0} \\ p_{\psi 0} \\ p_{\theta 0} \\ p_{\phi 0} \\ p_{\theta r0} \\ p_{\psi r0} \\ p_{zc0} \\ x_0 \\ y_0 \\ z_0 \\ \psi_0 \\ \theta_0 \\ \phi_0 \\ \theta_{r0} \\ \psi_{r0} \\ z_{c0} \end{bmatrix} = \begin{bmatrix} 360.502 \\ 0 \\ 142.916 \\ 0 \\ -148.867 \\ 0 \\ -121.824 \\ 0 \\ -10.542 \\ 0 \\ 0 \\ 0 \\ 0 \\ -0.1066 \\ 0 \\ 0.0753 \\ 0 \\ 2.14\times10^{-13} \end{bmatrix}. \tag{3.1.19}$$

The small perturbations were chosen as explained with Equation (3.1.15). The computed **F**, **G** and **W** matrices are given in Appendix B.

3.1.4 Comparison of Linear and Non-linear Model

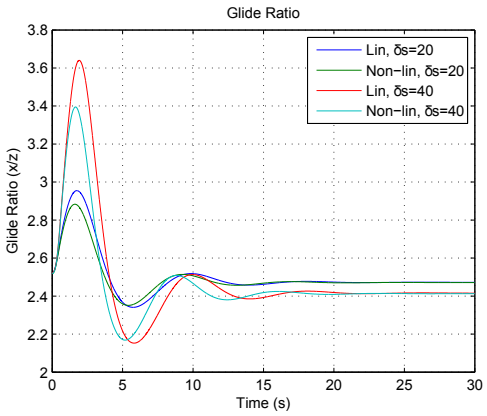
The linear model is now evaluated by comparing its response to the non-linear model. For homogeneous flight (zero brake deflection and wind), the output of the linear model closely matches the non-linear model. This is expected as homogeneous flight is the trim state about which the model was linearised. The linear model reflects the non-linear model for small deviations from the trim path. Symmetric brake deflection, δ_s, affects only the longitudinal states and Figure 3.1 compares the responses of the linear and non-linear models for two different magnitudes of δ_s step inputs. The linear model deviates increasingly for larger δ_s, showing a similar but less damped transient response than the non-linear model and a constant steady state error.

For asymmetric brake deflection, δ_a, the responses of the lateral states for the linear and non-linear model match well for small brake deflections, illustrated in Figure 3.2. For increasing δ_a, the transient response and frequency of oscillations change slightly and a steady state error is present. The linear model starts to deviate notably for δ_a larger than 30%, which is expected considering the increasing roll angle resulting from increasing δ_a, as seen in Figure 2.7(e) in Section 2.4.3.

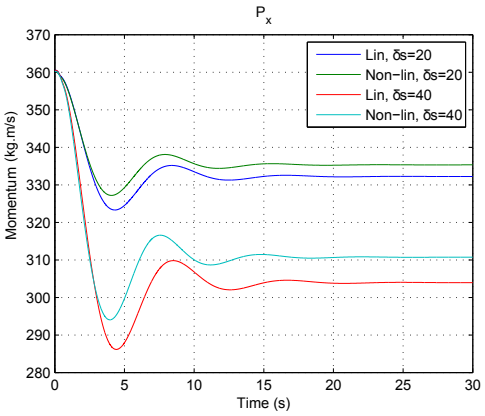
The linear model sufficiently describes the non-linear model for small brake deflections and even represents the yaw rate, shown in Figure 3.2(g), relatively well for larger brake deflections, which will be especially important for the flight control system design.

3.2 Analysis of Linear Dynamics

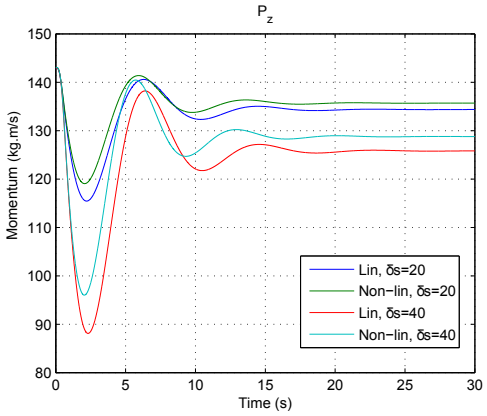
The linear model allows the natural modes of motion to be inspected in order to gain insight into the behaviour of the aircraft. The natural modes of motion are defined by the system poles, which govern the behaviour and motion of a linear system [21]. Recall that the linear



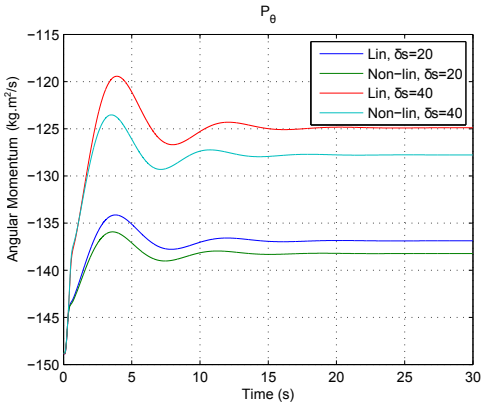
(a) Glide ratio



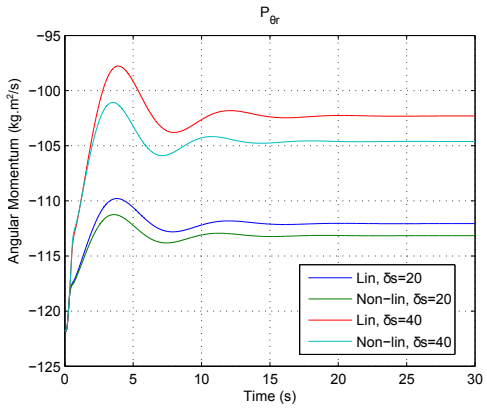
(b) P_x



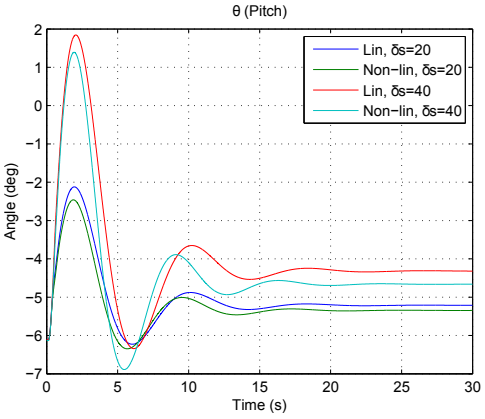
(c) P_z



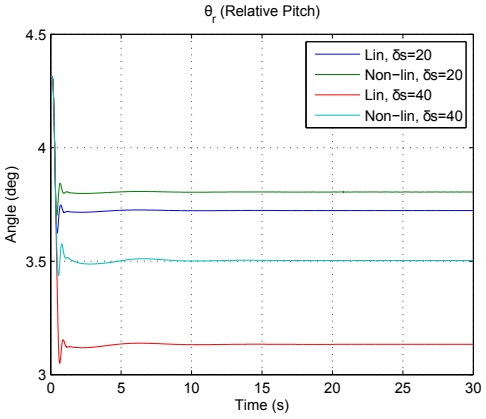
(d) P_θ



(e) $P_{\theta r}$

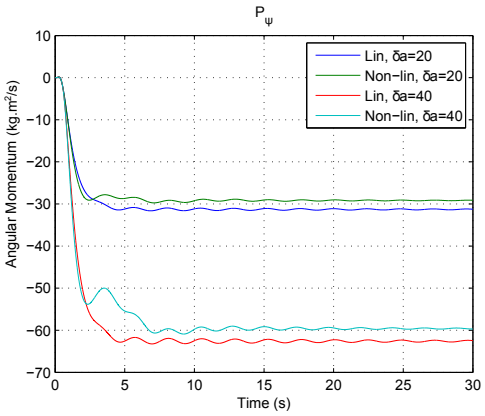


(f) θ (Pitch)

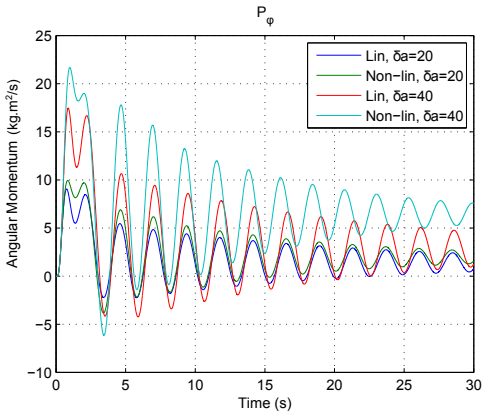


(g) θ_r (Relative pitch)

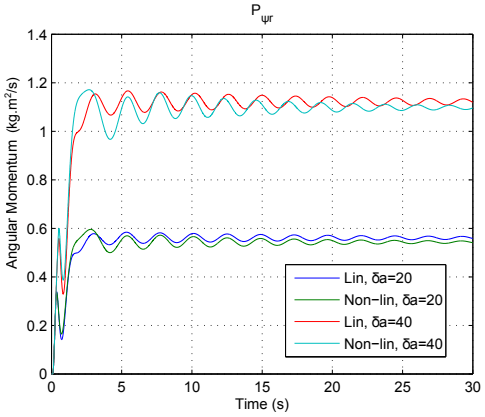
Figure 3.1 – Comparison of linear and non-linear simulation results for different δ_s step inputs.



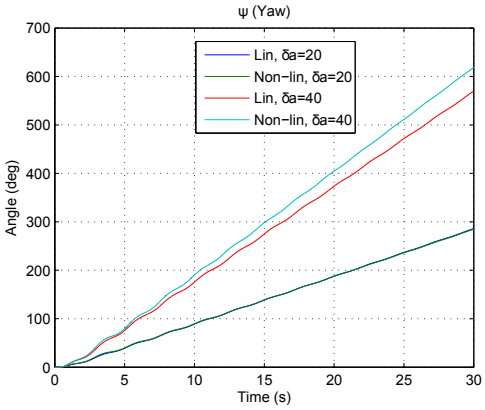
(a) P_ψ



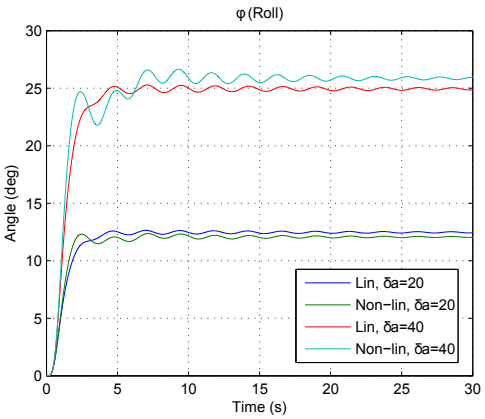
(b) P_ϕ



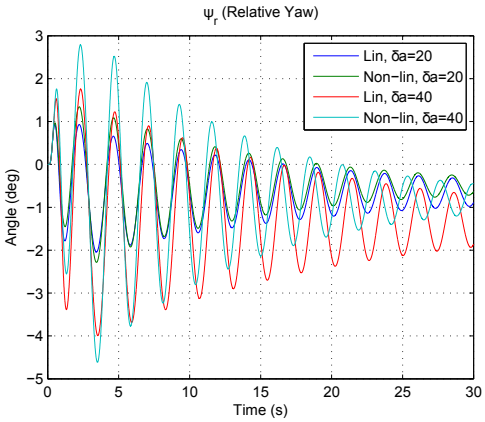
(c) $P_{\psi r}$



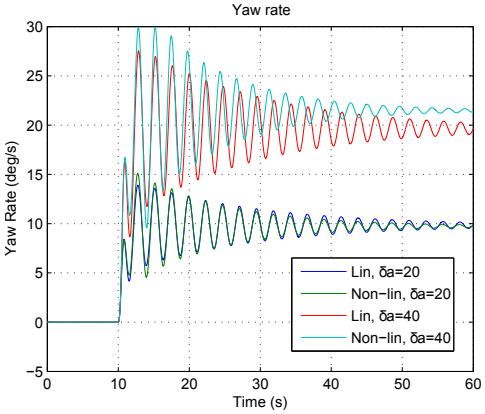
(d) ψ (Yaw)



(e) ϕ (Roll)



(f) ψ_r (Relative yaw)



(g) $\dot{\psi}$ (Yaw rate)

Figure 3.2 – Comparison of linear and non-linear simulation results for different δ_a step inputs.

model of the parafoil is represented by the following state space equations:

$$\dot{\mathbf{x}} = \mathbf{F}\mathbf{x} + \mathbf{G}\mathbf{u} + \mathbf{W}\mathbf{w} \tag{3.2.1}$$

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{D}\mathbf{u} \tag{3.2.2}$$

The poles can be found by solving the characteristic equation

$$\det(s\mathbf{I} - \mathbf{F}) = 0 \tag{3.2.3}$$

which is the same as calculating the eigenvalues of $\mathbf{F}$.

The poles are categorised as either longitudinal or lateral. The modes of motion mainly affect the states corresponding to one mode category (longitudinal or lateral) while the effects on the states from the other category are negligibly small. The parafoil is considered symmetrical about the XZ-plane, causing the longitudinal states to have no effect on, or no feedback to, the lateral states. Furthermore, the effect of the lateral states on the longitudinal states is small, especially when the deviations from trim are small. The lateral modes of motion tend to have a negligible small effect on the longitudinal states, and vice versa. The longitudinal related states and input vectors are

$$\mathbf{x}_{long} = \begin{bmatrix} p_x & p_z & p_\theta & p_{\theta r} & p_{zc} & x & z & \theta & \theta_r & z_c \end{bmatrix}^T \tag{3.2.4}$$

$$\mathbf{u}_{long} = \begin{bmatrix} \delta_s \end{bmatrix} \tag{3.2.5}$$

$$\mathbf{w}_{long} = \begin{bmatrix} w_x & w_z \end{bmatrix}^T \tag{3.2.6}$$

and the lateral are

$$\mathbf{x}_{lat} = \begin{bmatrix} p_y & p_\psi & p_\phi & p_{\psi r} & y & \psi & \phi & \psi_r \end{bmatrix}^T \tag{3.2.7}$$

$$\mathbf{u}_{lat} = \begin{bmatrix} \delta_a \end{bmatrix} \tag{3.2.8}$$

$$\mathbf{w}_{lat} = \begin{bmatrix} w_y \end{bmatrix} \tag{3.2.9}$$

The `eig` MATLAB function is used to calculate the eigenvalues. By analysing each pole or, in the case of a pair of complex conjugate eigenvalues, pole pair, information about the mode of motion it represents is gathered from the following:

- The position of the pole or pole pair is used to determine its natural damping and frequency.
- The modal form of the linear model is calculated from the state space equations [18, p. 155]. The modal form of Equation (3.2.1) is

  $$\dot{\mathbf{z}} = \mathbf{F}_m\mathbf{z} + \mathbf{G}_m\mathbf{u} + \mathbf{W}_m\mathbf{w} \tag{3.2.10}$$

  $$\mathbf{y}_m = \mathbf{H}_m\mathbf{z} + \mathbf{D}_m\mathbf{u} \tag{3.2.11}$$

  From the modal form of the input matrix, $\mathbf{G}_m$, a measure of the the controllability of a mode of motion can be determined [21].
- The modal form of the wind matrix, $\mathbf{W}_m$, gives a measure of how wind gust disturbances influence a mode of motion [21].
- The elements of $\mathbf{z}$ in Equation (3.2.10) are modal coordinates, each related to a mode of motion. The relationship between $\mathbf{z}$ and the state vector $\mathbf{x}$ is given by

  $$\mathbf{x} = \mathbf{V}'\mathbf{z} \tag{3.2.12}$$

  where $\mathbf{V}'$ is a matrix with the real and imaginary parts of the eigenvectors of $\mathbf{F}$ as its columns [21]. Thus, an eigenvector relates the states in $\mathbf{x}$ to a particular mode of motion and is used to quantify how much each state is influenced by a mode of motion.

- By applying the real parts of an eigenvector as a perturbation to the initial state vector of the linear simulation and observing the output response, the effects of the corresponding mode of motion can be graphically visualised.

It was stated that the elements of an eigenvector are used as a measure of the relative impact of a mode of motion on the different states. However, all of the states do not have similar units of magnitude. Thus, to quantify the relative effect a pole imposes on a given state variable due to the corresponding element in the eigenvector, the maximum variation experienced by the state variables during an arbitrary flight, in which typical, reasonable manoeuvres were performed, is determined and used to normalise the different rows of the eigenvector to represent the states with similar units of magnitude. The typical variation of the state variables is listed below in Equation (3.2.13). Position variables x, y and z are omitted since they do not feed back to any other state and do not contribute to the analysis. The resulting vector $\mathbf{x}_n$ contains the magnitudes for each state which will be used to normalise the eigenvectors.

$$\mathbf{x}_n = \begin{bmatrix} p_{xn} \\ p_{yn} \\ p_{zn} \\ p_{\psi n} \\ p_{\theta n} \\ p_{\phi n} \\ p_{\theta rn} \\ p_{\psi rn} \\ p_{zcn} \\ x_n \\ y_n \\ z_n \\ \psi_n \\ \theta_n \\ \phi_n \\ \theta_{rn} \\ \psi_{rn} \\ z_{cn} \end{bmatrix} = \begin{bmatrix} 350 \\ 350 \\ 40 \\ 65 \\ 17 \\ 20 \\ 10 \\ 1.1 \\ 0.5 \\ 1 \\ 1 \\ 1 \\ 0.08 \\ 0.08 \\ 0.08 \\ 0.009 \\ 0.05 \\ 5{\times}10^{-8} \end{bmatrix} = \begin{bmatrix} 350 \\ 350 \\ 40 \\ 65 \\ 17 \\ 20 \\ 10 \\ 1.1 \\ 0.5 \\ 1 \\ 1 \\ 1 \\ 4.58^\circ \\ 4.58^\circ \\ 4.58^\circ \\ 0.52^\circ \\ 2.86^\circ \\ 5{\times}10^{-8} \end{bmatrix} \tag{3.2.13}$$

Table 3.1 lists the relative magnitudes with which additional variables are scaled so that the effect on these variables due to the modes of motion can be represented with roughly similar units of magnitude. A similar approach to determining $\mathbf{x}_n$ was used and the typical variation from trim conditions are shown for each variable. For variables not listed in Table 3.1, unity scaling is appropriate while angles and rates are converted to degrees and deg/s respectively.

Table 3.2 lists the dominant linear and angular velocity contributions to each generalised momenta state. The dominant components were identified from the kinematic equations relating to the generalised momenta from [16], by making small angle approximations in order to eliminate the insignificant components. Since the generalised momenta states represent the momentum of the combined parafoil-payload system, some states contain multiple components. Components with the format V_{ab}^{I} refer to the velocity of system a (canopy or payload), in direction b with respect to inertial axes. From Table 3.2 it is evident that the components of interest are not each exclusively related to a single state. To ease analysis, the modal output equation,

$$\mathbf{y} = \mathbf{C}\mathbf{V}'\mathbf{z} + \mathbf{D}\mathbf{u} \tag{3.2.14}$$

Variable	Variation	Scale value
Angles of attack, α_A, α_L	2	10
Sideslip angles, β_A, β_L	6	2
Height (altitude), h	4	5
Forward velocity (body axes), u_A, u_L	2.7	10
Sideslip velocity (body axes), v_A, v_L	1.2	10
Payload downward velocity (body axes), w_L	0.16	100
Canopy downward velocity (body axes), w_A	0.47	50
Total velocity, V_A, V_L	2.7	10

Table 3.1 – Scaling of variables to represent similar units of magnitude.

p_x	p_y	p_z	p_ψ	p_θ	p_ϕ	$p_{\theta r}$	$p_{\psi r}$	p_{zc}
V^I_{px}				V^I_{px}		V^I_{px}		
V^I_{Lx}								
	V^I_{py}		V^I_{py}		V^I_{py}			
	V^I_{Ly}							
		V^I_{pz}		V^I_{pz}				V^I_{pz}
		V^I_{Lz}						
					p_p			
					p_L			
				q_p		q_p		
			q_L	q_L				
			r_p				r_p	
			r_L					

Table 3.2 – Dominant components for generalised momenta.

can be analysed similar to what was described for Equation (3.2.12). **C** and **D** are chosen to extract the following variables for longitudinal and lateral analysis, respectively:

$$\mathbf{y}_{long} = [u_L \; w_L \; q_L \; u_A \; w_A \; q_p \; \theta_L \; \theta_p]^T \tag{3.2.15}$$

$$\mathbf{y}_{lat} = [v_L \; p_L \; r_L \; v_A \; r_p \; \psi_L \; \phi_L \; \psi_p]^T \tag{3.2.16}$$

Canopy roll is omitted since the model considers it equal to the payload roll. Relative pitch and yaw states are replaced by canopy pitch and yaw. p_p is also omitted since, due to small angle approximations,

$$p_p \approx p_L \; . \tag{3.2.17}$$

In the following sections, the longitudinal and lateral poles are analysed.

3.2.1 Longitudinal Poles

The longitudinal model poles are listed below in Table 3.3 and plot in Figure 3.3 with exception of two additional integrators at the origin which have been omitted as they do not contribute to the analysis.

3.2.1.1 Longitudinal Complex Pair 1

The first complex conjugate pole pair is denoted by $\lambda_{4,5}$ and shown below with its natural frequency ω_n, damping ratio ζ and exponential decay time constant τ.

$$\lambda_{4,5} = -100 \pm 122.47j \; , \quad \begin{matrix} \omega_n \\ \\ \end{matrix} \begin{matrix} = 158.11 \text{ rad/s} \\ = 25.16 \text{ Hz} \end{matrix} \; , \quad \zeta = 0.63 \; , \quad \tau = 0.01 \text{ s} \tag{3.2.18}$$

Poles location	Denotation	Description
$-100 \pm 122.47j$	$\lambda_{4,5}$	Longitudinal complex pair 1
$-6.018 \pm 12.356j$	$\lambda_{6,7}$	Longitudinal complex pair 2
$-7.9345 \pm 4.8466j$	$\lambda_{8,9}$	Longitudinal complex pair 3
$-0.2723 \pm 0.7687j$	$\lambda_{17,18}$	Longitudinal complex pair 4

Table 3.3 – Longitudinal poles of the linear model.

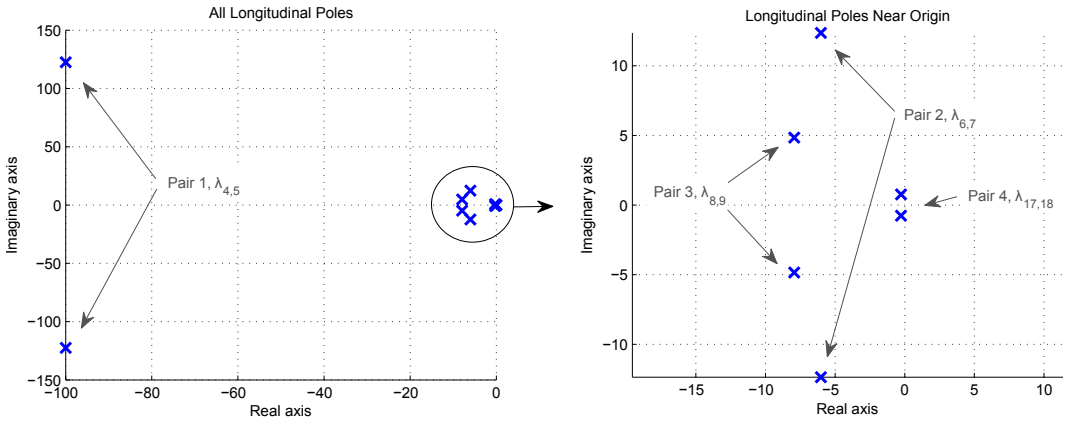


Figure 3.3 – Poles of the longitudinal model.

The eigenvector and modal output matrix column corresponding to the pole pair is shown in polar form below. The eigenvector, on the left-hand side, is scaled according to the elements of $\mathbf{x}_n$ in Equation (3.2.13) and normalised with the largest element in the eigenvector. Similarly, the column of the modal output matrix, on the right-hand side, is scaled with values listed in Table 3.1 and normalised.

$$\begin{bmatrix} p_x \\ p_z \\ p_\theta \\ p_{\theta r} \\ p_{zc} \\ \theta \\ \theta_r \\ z_c \end{bmatrix} = \begin{bmatrix} 9.9{\times}10^{-9}\ \angle 53^\circ \\ 2.1{\times}10^{-7}\ \angle 52^\circ \\ 6.5{\times}10^{-7}\ \angle 53^\circ \\ 8.6{\times}10^{-7}\ \angle 54^\circ \\ 3.3{\times}10^{-5}\ \angle -0^\circ \\ 8.1{\times}10^{-8}\ \angle 72^\circ \\ 7.6{\times}10^{-7}\ \angle 66^\circ \\ 1.0000\ \angle 49^\circ \end{bmatrix} \qquad \begin{bmatrix} u_L \\ w_L \\ q_L \\ u_A \\ w_A \\ q_p \\ \theta_L \\ \theta_p \end{bmatrix} = \begin{bmatrix} 0.0017\ \angle 14^\circ \\ 0.1402\ \angle -31^\circ \\ 0.1605\ \angle 21^\circ \\ 0.0472\ \angle -24^\circ \\ 1.0000\ \angle 2^\circ \\ 0.0191\ \angle -47^\circ \\ 0.0010\ \angle 72^\circ \\ 0.0001\ \angle 4^\circ \end{bmatrix} \tag{3.2.19}$$

As seen on the left hand side of Figure 3.3, this pole pair lies very far away from the rest of the system poles, with a very high relative frequency. The eigenvector indicates that this pole pair affects z_c several orders of magnitude more than the other states. p_{zc} is the most affected generalised momentum state and dominantly consists of vertical parafoil velocity w_A, which is seen to be the dominant output variable. z_c is related to the tension in the suspension lines. Furthermore, this pole pair is stable with a high frequency and is fairly well damped. From the part of the modal equation related to this pole pair, shown below, it is also seen that control and wind disturbance input do not notably affect this mode.

$$\begin{bmatrix} \dot{z}_4 \\ \dot{z}_5 \end{bmatrix} = \mathbf{F}_{m_{4,5}} \begin{bmatrix} z_4 \\ z_5 \end{bmatrix} + \mathbf{G}_{m_{4,5}} \begin{bmatrix} \delta_s \end{bmatrix} + \mathbf{W}_{m_{4,5}} \begin{bmatrix} w_x \\ w_z \end{bmatrix} \tag{3.2.20}$$

$$\mathbf{G}_{m_{4,5}} = \begin{bmatrix} 6.09{\times}10^{-7} \\ 1.62{\times}10^{-5} \end{bmatrix} \qquad \mathbf{W}_{m_{4,5}} = \begin{bmatrix} -2.29{\times}10^{-5} & 6.78{\times}10^{-5} \\ -2.41{\times}10^{-4} & 5.4{\times}10^{-4} \end{bmatrix} \tag{3.2.21}$$

As such, this pole pair is not important for the analysis of the movement of the system and does not warrant any further investigation for the scope of this analysis.

3.2.1.2 Longitudinal Complex Pair 2

The second longitudinal complex conjugate pole pair is described as follows.

$$\lambda_{6,7} = -6.018 \pm 12.356j \ , \qquad \omega_n \begin{matrix} = 13.74 \text{ rad/s} \\ = 2.19 \text{ Hz} \end{matrix} \ , \qquad \zeta = 0.44 \ , \qquad \tau = 0.17 \text{ s} \tag{3.2.22}$$

Similar to before, the scaled and normalised eigenvector and output vector related to this pole pair is given below.

$$\begin{bmatrix} p_x \\ p_z \\ p_\theta \\ p_{\theta r} \\ p_{zc} \\ \theta \\ \theta_r \\ z_c \end{bmatrix} = \begin{bmatrix} 0.0004 \angle{-4}^\circ \\ 0.0087 \angle{-29}^\circ \\ 0.0285 \angle{-0}^\circ \\ 0.0378 \angle{-16}^\circ \\ 0.0912 \angle{47}^\circ \\ 0.0888 \angle{-14}^\circ \\ 1.0000 \angle{-9}^\circ \\ 0.5126 \angle{-77}^\circ \end{bmatrix} \qquad \begin{bmatrix} u_L \\ w_L \\ q_L \\ u_A \\ w_A \\ q_p \\ \theta_L \\ \theta_p \end{bmatrix} = \begin{bmatrix} 0.0347 \angle{-13}^\circ \\ 1.0000 \angle{-16}^\circ \\ 0.5353 \angle{-78}^\circ \\ 0.0339 \angle{-14}^\circ \\ 0.0533 \angle{-90}^\circ \\ 0.1521 \angle{-56}^\circ \\ 0.0389 \angle{-14}^\circ \\ 0.0111 \angle{8}^\circ \end{bmatrix} \tag{3.2.23}$$

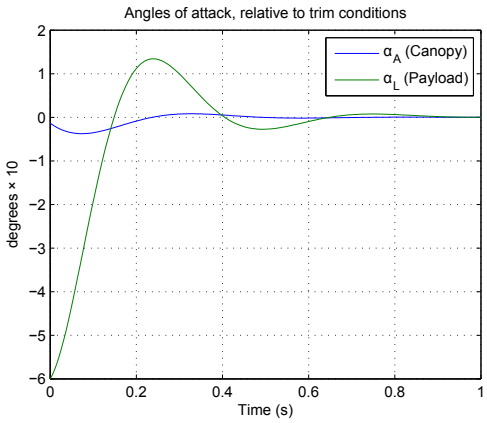
This pole pair has a relatively high frequency compared to the other longitudinal pole pairs (ignoring the first pole pair for reasons explained above) and mainly contributes to w_L (thus also payload angle of attack α_L), q_L to some extent and q_p to a lesser extent. The response of the mode can be observed in simulation by initialising the state vector to the real parts of the eigenvector related to this mode. The results are plotted in Figure 3.4. Note that the angle of attack and velocity responses are scaled by factors indicated in Table 3.1 for the magnitudes of the responses to be better comparable. The results show high magnitude damped oscillations for payload angle of attack and pitch rate. This mode is described as the longitudinal payload pendulum mode by [22] in which a pendulum type of swinging motion exists between the canopy and payload. Despite the large oscillation magnitudes in the payload variables, the motion dies out relatively quickly after approximately one oscillation period and does not notably impact the flight path.

The part of the modal equation related to this mode is shown in Equation (3.2.24) and the modal input matrix $\mathbf{G}_{m_{6,7}}$ and wind matrix $\mathbf{W}_{m_{6,7}}$ terms are expanded in Equation (3.2.25). Vertical wind has the dominating effect, as seen in the second column of $\mathbf{W}_{m_{6,7}}$.

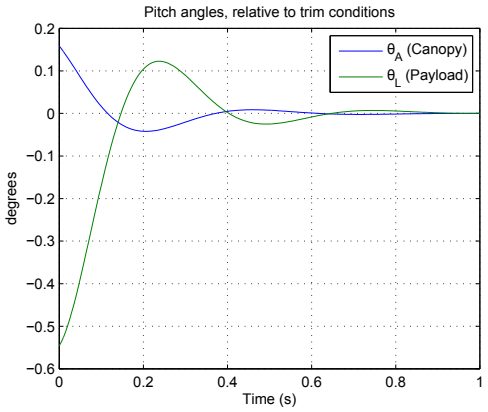
$$\begin{bmatrix} \dot{z}_6 \\ \dot{z}_7 \end{bmatrix} = \mathbf{F}_{m_{6,7}} \begin{bmatrix} z_6 \\ z_7 \end{bmatrix} + \mathbf{G}_{m_{6,7}} \begin{bmatrix} \delta_s \end{bmatrix} + \mathbf{W}_{m_{6,7}} \begin{bmatrix} w_x \\ w_z \end{bmatrix} \tag{3.2.24}$$

$$\mathbf{G}_{m_{6,7}} = \begin{bmatrix} -0.2 \\ -0.66 \end{bmatrix} \qquad \mathbf{W}_{m_{6,7}} = \begin{bmatrix} 3.23 & -5.6 \\ -1.32 & 10.95 \end{bmatrix} \tag{3.2.25}$$

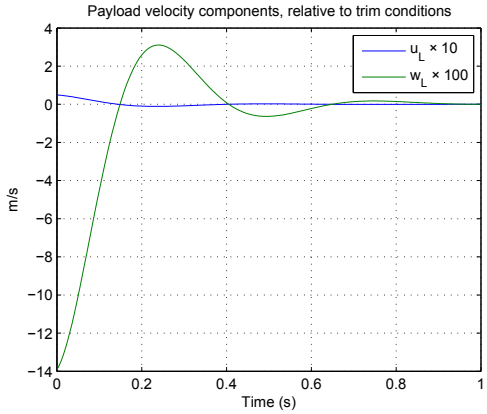
Being adequately damped with a short period, this mode is not troublesome regarding the stability or usefulness of the system in the scope of FCS design.



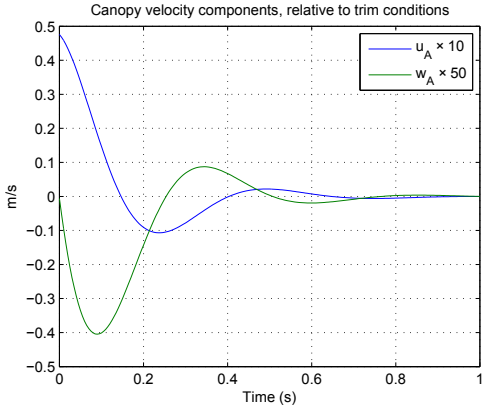
(a) Angles of attack



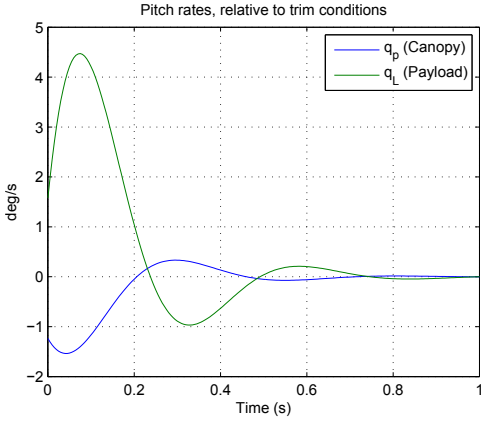
(b) Pitch angles



(c) Payload velocities



(d) Canopy velocities



(e) Pitch rates

Figure 3.4 – Longitudinal complex pole pair 2 ($\lambda_{6,7}$) response.

3.2.1.3 Longitudinal Complex Pair 3

The third longitudinal complex pole pair is described as follows.

$$\lambda_{8,9} = -7.9345 \pm 4.8466j \ , \qquad \begin{matrix} \omega_n \\ \\ \end{matrix} \begin{matrix} = 9.2976 \text{ rad/s} \\ = 1.4798 \text{ Hz} \end{matrix} \ , \quad \zeta = 0.85 \ , \quad \tau = 0.13 \text{ s} \tag{3.2.26}$$

The scaled and normalised eigenvector and modal output vector related to this mode is given below in polar form.

$$\begin{bmatrix} p_x \\ p_z \\ p_\theta \\ p_{\theta r} \\ p_{zc} \\ \theta \\ \theta_r \\ z_c \end{bmatrix} = \begin{bmatrix} 0.0090 \angle 6° \\ 0.1313 \angle -19° \\ 0.6675 \angle 0° \\ 0.9091 \angle -0° \\ 1.0000 \angle 76° \\ 0.3342 \angle 30° \\ 0.7155 \angle 12° \\ 0.2442 \angle 21° \end{bmatrix} \qquad \begin{bmatrix} u_L \\ w_L \\ q_L \\ u_A \\ w_A \\ q_p \\ \theta_L \\ \theta_p \end{bmatrix} = \begin{bmatrix} 0.0461 \angle 26° \\ 1.0000 \angle 64° \\ 0.5447 \angle -2° \\ 0.3480 \angle 1° \\ 0.6862 \angle -34° \\ 0.4213 \angle 4° \\ 0.0586 \angle 30° \\ 0.0453 \angle 35° \end{bmatrix} \tag{3.2.27}$$

This pole pair is dominant in w_L, w_A, q_L, q_p and u_A. When initialising the state vector with the real part of the eigenvector related to this mode like before, the simulation results, plotted in Figure 3.5, show a damped response in pitch rate and velocity.

Conventional aircraft posses a short period mode which is a damped oscillation in pitch about the y axis of the aircraft [23]. The short period mode is usually excited when the equilibrium pitch state of the aircraft is disturbed. The damping effect of the oscillation is caused by the tendency of the wing to align with incident air flow. The principle variables of the mode is angle of attack, pitch rate and pitch attitude [23] and the mode features an approximately constant velocity due to inertia and momentum effects preventing a notable velocity response in the short time frame of the mode [23].

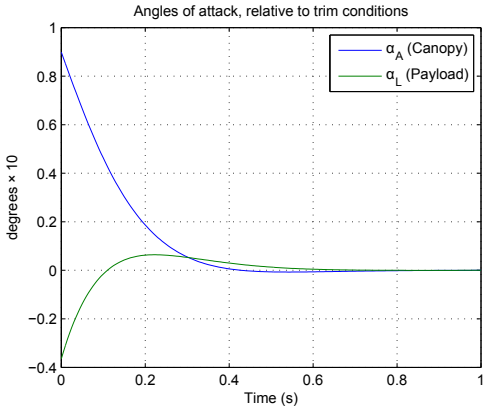
The normal payload velocity shows a damped response while the axial velocity changes relatively little, in a typical short period fashion. The canopy also experiences a damped angle of attack response, but with a comparable axial velocity change, as opposed to expected from classical short period modes. A relatively large but well damped pitch rate deviation is present and the pitch angles quickly settle to the trim values.

Equation (3.2.29) shows that both longitudinal wind components can play a major roll in the excitation of this mode with some controllability by δ_s.

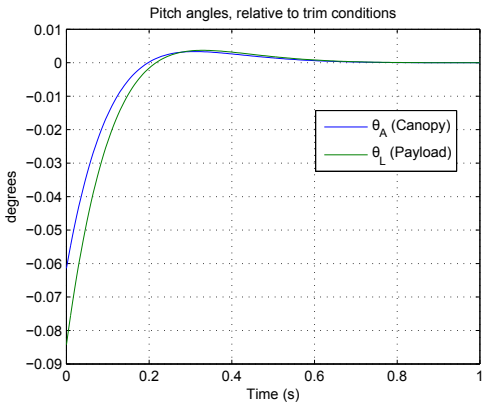
$$\begin{bmatrix} \dot{z}_8 \\ \dot{z}_9 \end{bmatrix} = \mathbf{F}_{m_{8,9}} \begin{bmatrix} z_8 \\ z_9 \end{bmatrix} + \mathbf{G}_{m_{8,9}} \begin{bmatrix} \delta_s \end{bmatrix} + \mathbf{W}_{m_{8,9}} \begin{bmatrix} w_x \\ w_z \end{bmatrix} \tag{3.2.28}$$

$$\mathbf{G}_{m_{8,9}} = \begin{bmatrix} 2.82 \\ -0.22 \end{bmatrix} \qquad \mathbf{W}_{m_{8,9}} = \begin{bmatrix} -131.83 & 346.28 \\ 113.26 & -536.19 \end{bmatrix} \tag{3.2.29}$$

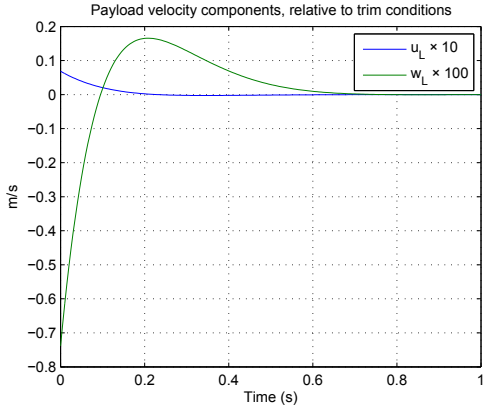
Observed from the payload, the motion of this mode mimics a conventional short period mode, although the canopy experiences a comparable axial velocity change. This mode is stable, very well damped and with a sufficiently short time span presents no adverse effects in the context of autonomous control.



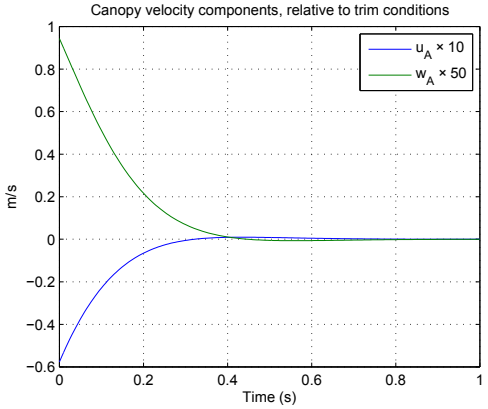
(a) Angles of attack



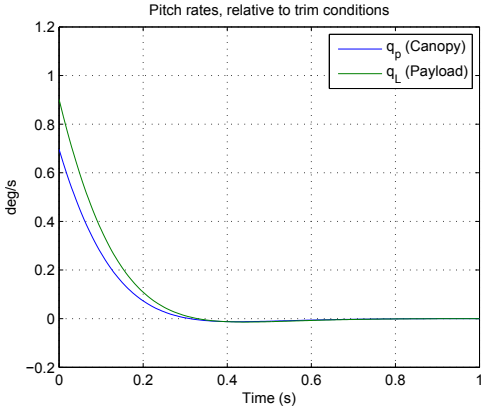
(b) Pitch angles



(c) Payload velocities



(d) Canopy velocities



(e) Pitch rates

Figure 3.5 – Longitudinal complex pole pair 3 ($\lambda_{8,9}$) response.

3.2.1.4 Longitudinal Complex Pair 4

Longitudinal complex pole pair 4 is described as follows.

$$\lambda_{17,18} = -0.2723 \pm 0.7687j \ , \qquad \omega_n \begin{matrix} = 0.82 \text{ rad/s} \\ = 0.13 \text{ Hz} \end{matrix} \ , \quad \zeta = 0.33 \ , \quad \tau = 3.67 \text{ s} \tag{3.2.30}$$

The scaled and normalised eigenvector and modal output vector related to this mode is given below in polar form.

$$\begin{bmatrix} p_x \\ p_z \\ p_\theta \\ p_{\theta r} \\ p_{zc} \\ \theta \\ \theta_r \\ z_c \end{bmatrix} = \begin{bmatrix} 0.0720 \angle 87° \\ 0.7531 \angle -0° \\ 0.3967 \angle -70° \\ 0.5590 \angle -70° \\ 0.8319 \angle -72° \\ 1.0000 \angle 12° \\ 0.0229 \angle 2° \\ 0.2049 \angle -66° \end{bmatrix} \qquad \begin{bmatrix} u_L \\ w_L \\ q_L \\ u_A \\ w_A \\ q_p \\ \theta_L \\ \theta_p \end{bmatrix} = \begin{bmatrix} 0.7606 \angle -60° \\ 1.0000 \angle -58° \\ 0.2909 \angle -59° \\ 0.6100 \angle -46° \\ 0.5081 \angle -2° \\ 0.2902 \angle -59° \\ 0.3567 \angle 12° \\ 0.3558 \angle 12° \end{bmatrix} \tag{3.2.31}$$

This poorly damped pole pair has a low frequency relative to the other longitudinal pole pairs. The normalised modal output vector shows that velocities w_L, u_L, u_A and w_A are dominant. Simulation results, where the initial state vector is set to the real parts of the eigenvector, are plotted in Figure 3.6 and show dominant oscillations in both axial and normal velocity. The oscillation causes the parafoil to fly a damped oscillatory flight path relative to the trim glide slope as seen in the height graph.

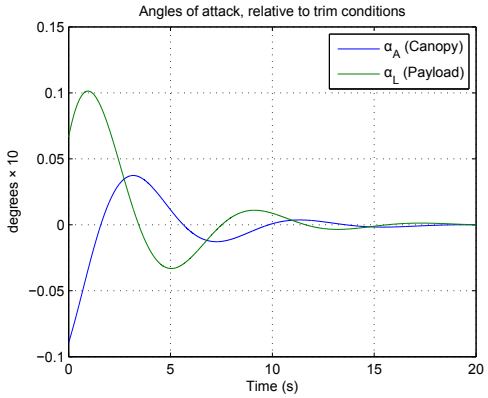
This response is similar to that of the phugoid mode witnessed in conventional aircraft where a disturbance in forward velocity causes a lightly damped low frequency oscillation which couples into pitch attitude and height [23]. The motion is explained by [23]: while flying at equilibrium trim conditions, the aircraft might experience a disturbance that slightly reduces forward velocity while the angle of attack remains virtually constant. This leads to a reduction in lift causing the aircraft to lose height and fly downward. In turn, this causes an increase in speed and thus lift, growing until the aircraft stops losing height, steadily pitches upward and starts climbing again. The aircraft now has an excess of kinetic energy and flies up, through the nominal trim path datum, decelerating and gently starts to pitch down. The lift decreases and eventually the process repeats itself. The oscillation is damped by effects of drag and eventually dies away. In conventional aircraft which are usually designed with low drag, this mode is, when stable, often poorly damped and the oscillation may continue for many cycles before eventually being damped out [23]. However, in this case, the pole pair is lightly damped and simulation results in Figure 3.6 show that the motion dies out in about 2 periods. The current case differs somewhat from the conventional case in that the normal velocities actually change notably meaning that the angles of attack are not considered constant related to the other states.

Similar to before, the effects of δ_s and longitudinal wind components are given in Equation (3.2.33). δ_s has some control with respect to this mode while both longitudinal wind components have a greater effect.

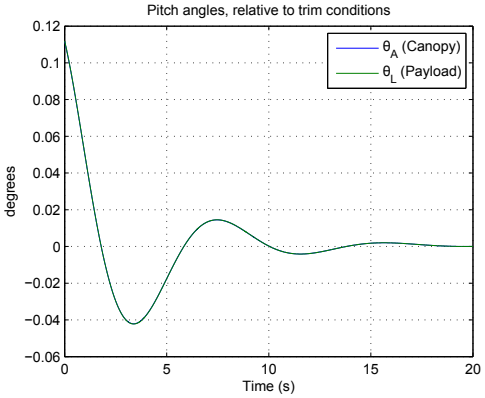
$$\begin{bmatrix} \dot{z}_{17} \\ \dot{z}_{18} \end{bmatrix} = \mathbf{F}_{m_{17,18}} \begin{bmatrix} z_{17} \\ z_{18} \end{bmatrix} + \mathbf{G}_{m_{17,18}} \begin{bmatrix} \delta_s \end{bmatrix} + \mathbf{W}_{m_{17,18}} \begin{bmatrix} w_x \\ w_z \end{bmatrix} \tag{3.2.32}$$

$$\mathbf{G}_{m_{17,18}} = \begin{bmatrix} 0.5 \\ 0.24 \end{bmatrix} \qquad \mathbf{W}_{m_{17,18}} = \begin{bmatrix} -34.88 & -6.63 \\ 7.91 & -20.51 \end{bmatrix} \tag{3.2.33}$$

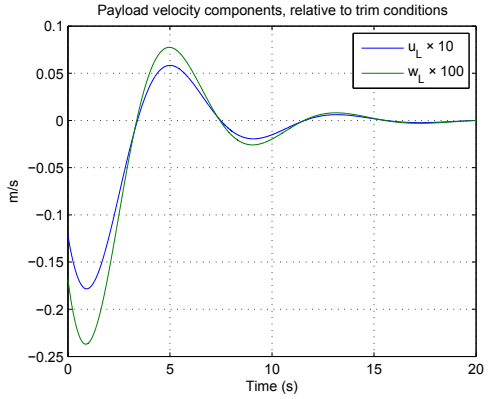
An additional factor which may potentially excite the phugoid mode is the launch of the parafoil with the UCT launcher system. It has been observed that the parafoil flies slightly upward when launched, possibly related to the launch speed, after which it reaches a maximum height and starts descending, performing a phugoid-like motion.



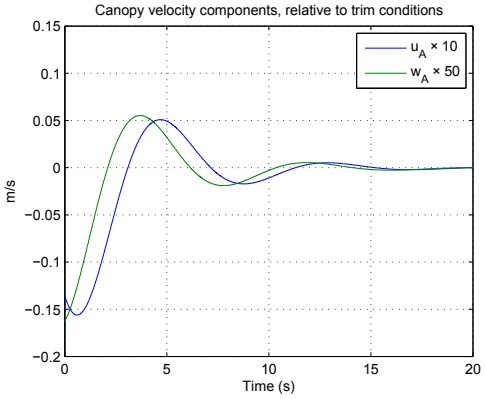
(a) Angles of attack



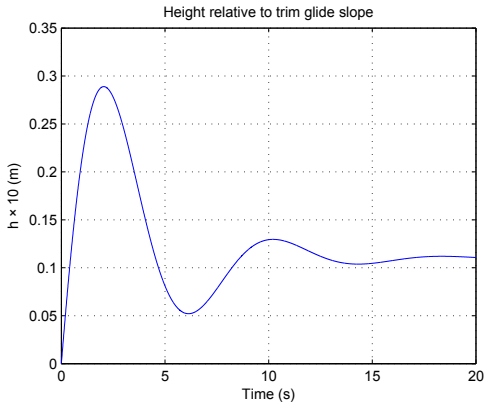
(b) Pitch angles



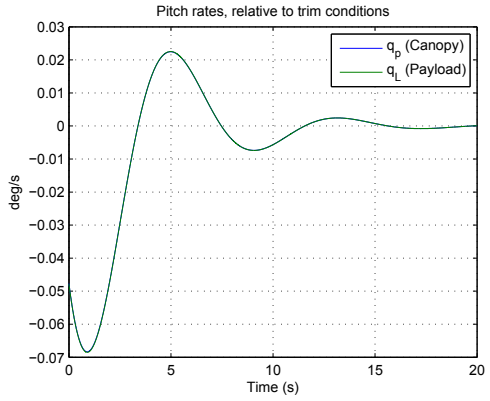
(c) Payload velocities



(d) Canopy velocities



(e) Height



(f) Pitch rates

Figure 3.6 – Longitudinal complex pole pair 4 ($\lambda_{17,18}$) response: phugoid mode.

3.2.2 Lateral Poles

The lateral model poles are listed below in Table 3.4 and plotted in Figure 3.7. Two additional integrators related to the ψ and y states have been omitted from the analysis.

Poles location	Denotation	Description
$-2.9603 \pm 4.363j$	$\lambda_{10,11}$	Lateral complex pair 1
-4.3203	λ_{12}	Lateral real pole 1
$-0.0562 \pm 2.6301j$	$\lambda_{13,14}$	Lateral complex pair 2
-1.2051	λ_{15}	Lateral real pole 2

Table 3.4 – Lateral poles of the linear model.

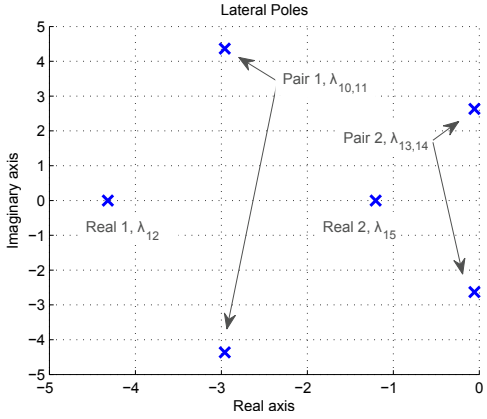


Figure 3.7 – Poles of the lateral model.

3.2.2.1 Lateral Complex Pair 1

The first lateral complex pole pair is described as follows.

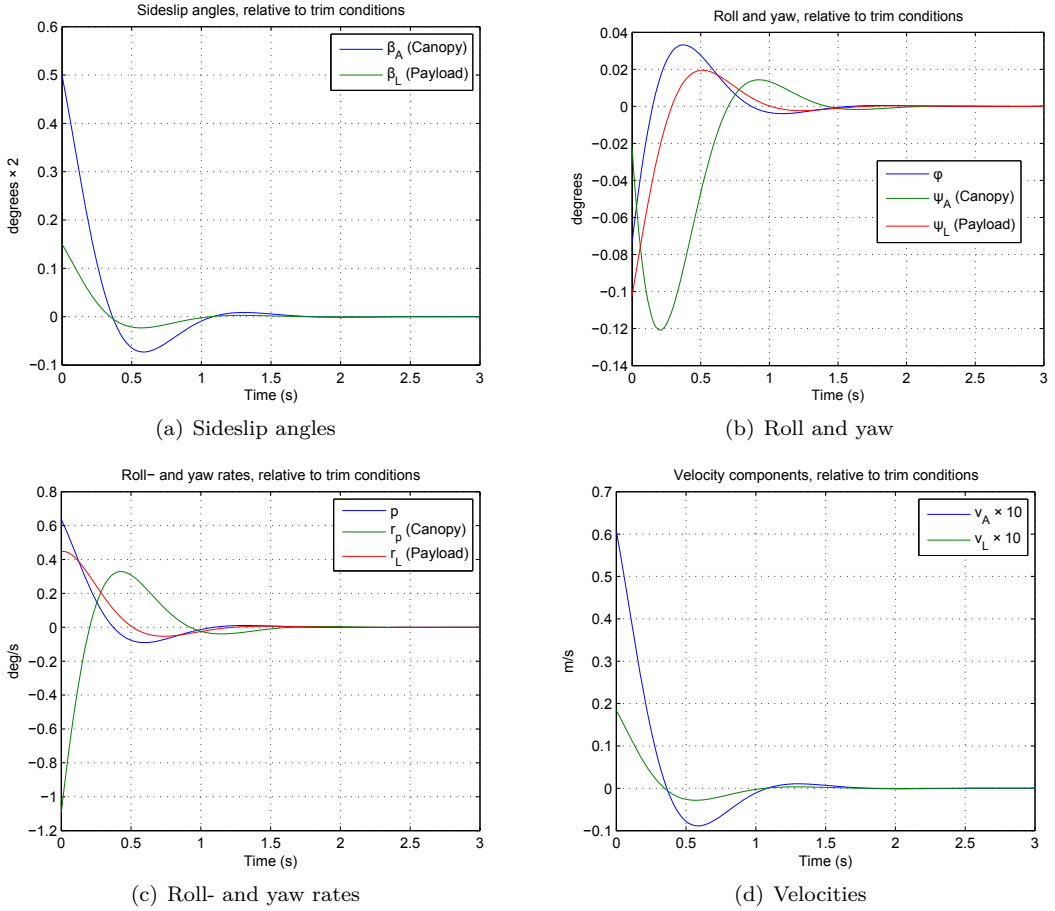
$$\lambda_{10,11} = -2.9603 \pm 4.363j \ , \qquad \begin{matrix} \omega_n \\ \\ \end{matrix} \begin{matrix} = 5.27 \text{ rad/s} \\ = 0.84 \text{ Hz} \end{matrix} \ , \qquad \zeta = 0.56 \ , \qquad \tau = 0.34 \text{ s} \tag{3.2.34}$$

The scaled and normalised eigenvector and modal output vector related to this mode is given below.

$$\begin{bmatrix} p_y \\ p_\psi \\ p_\phi \\ p_{\psi r} \\ \psi \\ \phi \\ \psi_r \end{bmatrix} = \begin{bmatrix} 0.0061 \angle 26° \\ 0.0374 \angle 45° \\ 0.4320 \angle 0° \\ 0.6599 \angle 40° \\ 0.2170 \angle 18° \\ 0.2445 \angle 53° \\ 1.0000 \angle 75° \end{bmatrix} \qquad \begin{bmatrix} v_L \\ p_L \\ r_L \\ v_A \\ r_p \\ \psi_L \\ \phi_L \\ \psi_p \end{bmatrix} = \begin{bmatrix} 0.1308 \angle 4° \\ 0.4531 \angle -3° \\ 0.4020 \angle -38° \\ 0.4313 \angle -0° \\ 1.0000 \angle 39° \\ 0.0762 \angle 18° \\ 0.0859 \angle 53° \\ 0.1897 \angle -85° \end{bmatrix} \tag{3.2.35}$$

This pole pair is moderately damped and is dominant in canopy yaw rate r_p, to a lesser extent in p_L, r_L, v_A and less in v_L and ψ_p. Figure 3.8 shows the simulation results where the state vector was initialised with the real parts of the eigenvector. A moderately damped

yaw and rolling oscillation is present, appearing similar to a dutch roll type of motion as encountered in classical fixed wing aircraft.



(a) Sideslip angles

(b) Roll and yaw

(c) Roll- and yaw rates

(d) Velocities

Figure 3.8 – Lateral complex pole pair 1 ($\lambda_{10,11}$) response.

The dutch roll is a damped oscillation in yaw which couples into roll and also sideslip to a lesser extent [23]. In a conventional fixed wing aircraft, a disturbance in yaw leads to an oscillation due to a restoring yawing moment, largely due to the fin [23]. The yawing motion causes the relative velocity over the port and starboard wing to vary, causing differential lift and drag perturbations leading to a roll oscillation, lagging the yaw motion by approximately 90° [23]. These oscillations cause the so-called dutch roll motion where the aircraft yaws and rolls such that the forward-moving wing is low and the aft-moving wing is high. Usually, for a stable dutch roll mode the peak roll to peak yaw ratio is less than one.

Since the parafoil does not have a fin to contribute to the dutch roll motion, the motion is caused by other aerodynamic effects. Cook [23] mentions that other aerodynamic effects are also present for a conventional aircraft, but also notes that it is very difficult to quantify all the aerodynamic contributions to the mode with any degree of confidence. Lingard [7] notes that the dutch roll mode is usually lightly damped for large parafoils but that shorter suspension lines and an increased mass ratio tend to stabilise the mode. The peak roll to peak yaw ratio is larger than what would be expected from a classical dutch roll mode. Nevertheless, the mode is moderately well damped and has a relatively high frequency in comparison to the other lateral modes, dying away in less than 2 seconds. The payload experiences

relatively little β_L and r_L perturbation, with the total velocity remaining approximately constant.

An extract of the modal form of the linear model related to this mode of motion is given below. This mode can be controlled by asymmetric deflection and very easily be induced by cross-wind disturbances.

$$\begin{bmatrix} \dot{z}_{10} \\ \dot{z}_{11} \end{bmatrix} = \mathbf{F}_{m_{10,11}} \begin{bmatrix} z_{10} \\ z_{11} \end{bmatrix} + \mathbf{G}_{m_{10,11}} \begin{bmatrix} \delta_a \end{bmatrix} + \mathbf{W}_{m_{10,11}} \begin{bmatrix} w_y \end{bmatrix} \tag{3.2.36}$$

$$\mathbf{G}_{m_{10,11}} = \begin{bmatrix} 2.18 \\ -4.6 \end{bmatrix} \qquad \mathbf{W}_{m_{10,11}} = \begin{bmatrix} 121.47 \\ -62.19 \end{bmatrix} \tag{3.2.37}$$

3.2.2.2 Lateral Complex Pair 2

The second lateral complex pole pair is described as follows.

$$\lambda_{13,14} = -0.0562 \pm 2.6301j \ , \quad \begin{matrix} \omega_n & = 2.63 \text{ rad/s} \\ & = 0.69 \text{ Hz} \end{matrix} \ , \quad \zeta = 0.02 \ , \quad \tau = 17.79 \text{ s} \tag{3.2.38}$$

The scaled and normalised eigenvector and modal output vector related to this mode is given below.

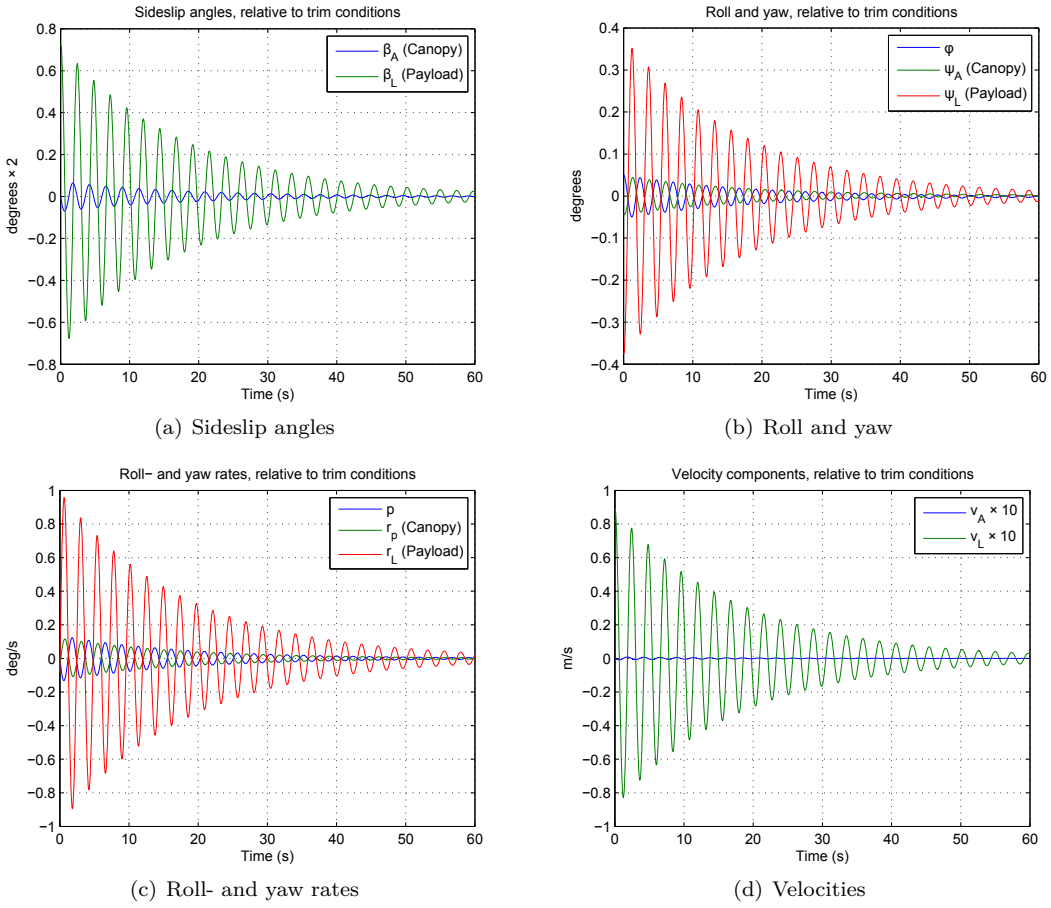
$$\begin{bmatrix} p_y \\ p_\psi \\ p_\phi \\ p_{\psi r} \\ \psi \\ \phi \\ \psi_r \end{bmatrix} = \begin{bmatrix} 0.0008 \angle 68° \\ 0.0136 \angle 20° \\ 0.4279 \angle -0° \\ 0.0495 \angle 68° \\ 0.7079 \angle -6° \\ 0.0975 \angle -1° \\ 1.0000 \angle -4° \end{bmatrix} \qquad \begin{bmatrix} v_L \\ p_L \\ r_L \\ v_A \\ r_p \\ \psi_L \\ \phi_L \\ \psi_p \end{bmatrix} = \begin{bmatrix} 0.8966 \angle -6° \\ 0.1377 \angle -89° \\ 1.0000 \angle 85° \\ 0.0881 \angle 90° \\ 0.1219 \angle 70° \\ 0.3801 \angle -6° \\ 0.0524 \angle -1° \\ 0.0463 \angle -21° \end{bmatrix} \tag{3.2.39}$$

This pole pair is extremely poorly damped with a dominant effect on r_L and v_L, and ψ_L, p_L and r_p to a smaller extent. When initialising the state vector with the real parts of the eigenvector related to this mode, simulation results in Figure 3.9 show badly damped oscillations in payload yaw, yaw rate and sideslip, with relatively small magnitude rolling and canopy effects, persisting for more than 60 s. This is the payload relative twist mode of motion mentioned in Section 1.2.1 and also seen in Section 2.4.3 when an asymmetric brake deflection step was applied. The setup and interaction of the suspension lines between the canopy and payload box result in a line twist moment which can be modeled as a non-linear rotational spring and damper system [5] which is very badly damped in this case. The oscillation is coupled to the canopy which experiences oscillations of a smaller magnitude. In Section 2.4.3 it was noted that for a winged UAV payload, aerodynamic damping reduces the oscillation persistence. Since the current payload has no such means of damping, alternate strategies will have to be explored. The oscillation has negligibly little effect on inertial velocity, but with sensors and instrumentation on-board the payload, it is desirable to prevent the excitation of this mode.

The modal form of the linear equations related to this mode in Equations (3.2.40) and (3.2.41) show controllability, although relatively low in comparison to the other lateral modes, and that the mode can also be induced by a cross-wind disturbance.

$$\begin{bmatrix} \dot{z}_{13} \\ \dot{z}_{14} \end{bmatrix} = \mathbf{F}_{m_{13,14}} \begin{bmatrix} z_{13} \\ z_{14} \end{bmatrix} + \mathbf{G}_{m_{13,14}} \begin{bmatrix} \delta_a \end{bmatrix} + \mathbf{W}_{m_{13,14}} \begin{bmatrix} w_y \end{bmatrix} \tag{3.2.40}$$

$$\mathbf{G}_{m_{13,14}} = \begin{bmatrix} 0.56 \\ 0.37 \end{bmatrix} \qquad \mathbf{W}_{m_{13,14}} = \begin{bmatrix} -5.09 \\ 36.97 \end{bmatrix} \tag{3.2.41}$$



(a) Sideslip angles

(b) Roll and yaw

(c) Roll- and yaw rates

(d) Velocities

Figure 3.9 – Lateral complex pole pair 2 ($\lambda_{13,14}$) response: payload twist oscillation.

3.2.2.3 Lateral Real Pole 1 and 2

The remaining two stable real lateral poles are real pole 1, described by

$$\lambda_{12} = -4.3203\ , \qquad \tau = 0.23\ \text{s} \tag{3.2.42}$$

and real pole 2, described by

$$\lambda_{15} = -1.2051\ , \qquad \tau = 0.83\ \text{s}\ . \tag{3.2.43}$$

The scaled and normalised eigenvectors and modal output vectors related to these modes are given below.

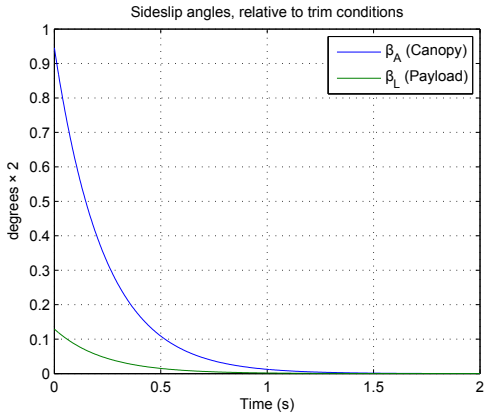
$$\lambda_{12}: \quad \begin{bmatrix} p_y \\ p_\psi \\ p_\phi \\ p_{\psi r} \\ \psi \\ \phi \\ \psi_r \end{bmatrix} = \begin{bmatrix} 0.0162\ \angle 180° \\ 0.1246\ \angle 180° \\ 0.7283\ \angle 180° \\ 0.9875\ \angle 180° \\ 0.2969\ \angle 0° \\ 0.5236\ \angle 0° \\ 1.0000\ \angle 0° \end{bmatrix} \qquad \begin{bmatrix} v_L \\ p_L \\ r_L \\ v_A \\ r_p \\ \psi_L \\ \phi_L \\ \psi_p \end{bmatrix} = \begin{bmatrix} 0.1312\ \angle 180° \\ 0.4849\ \angle 180° \\ 0.2749\ \angle 180° \\ 1.0000\ \angle 180° \\ 0.8536\ \angle 180° \\ 0.0636\ \angle 0° \\ 0.1122\ \angle 0° \\ 0.1976\ \angle 0° \end{bmatrix} \tag{3.2.44}$$

$$\lambda_{15}: \quad \begin{bmatrix} p_y \\ p_\psi \\ p_\phi \\ p_{\psi r} \\ \psi \\ \phi \\ \psi_r \end{bmatrix} = \begin{bmatrix} 0.0555\ \angle 0° \\ 0.1849\ \angle 0° \\ 0.3404\ \angle 0° \\ 0.0666\ \angle 180° \\ 0.2626\ \angle 0° \\ 1.0000\ \angle 180° \\ 0.0720\ \angle 0° \end{bmatrix} \qquad \begin{bmatrix} v_L \\ p_L \\ r_L \\ v_A \\ r_p \\ \psi_L \\ \phi_L \\ \psi_p \end{bmatrix} = \begin{bmatrix} 0.1554\ \angle 0° \\ 1.0000\ \angle 0° \\ 0.2626\ \angle 180° \\ 0.4244\ \angle 0° \\ 0.3076\ \angle 180° \\ 0.2179\ \angle 0° \\ 0.8298\ \angle 180° \\ 0.2553\ \angle 0° \end{bmatrix} \tag{3.2.45}$$

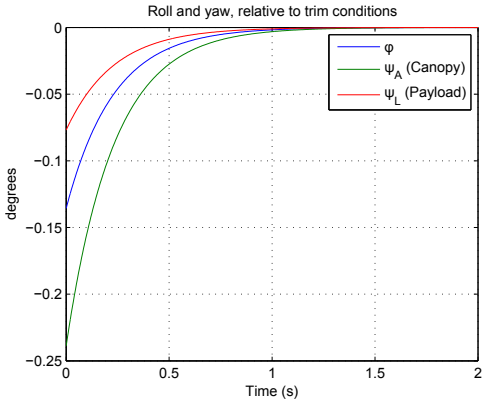
λ_{12} contributes mainly to v_A and r_p, with some contribution also in p_L, r_L, and relatively little in v_L, ϕ_L and ψ_p.

λ_{15} is dominant in p_L and ϕ_L with relative less, but notable contribution to the other variables.

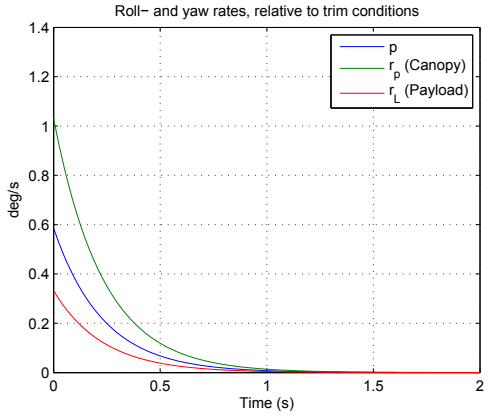
Figures 3.10 and 3.11 show the simulation results for the response of each mode, having initialised the state vector to the corresponding eigenvectors.



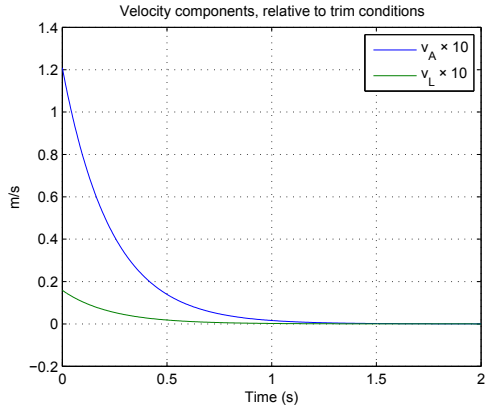
(a) Sideslip angles



(b) Roll and yaw



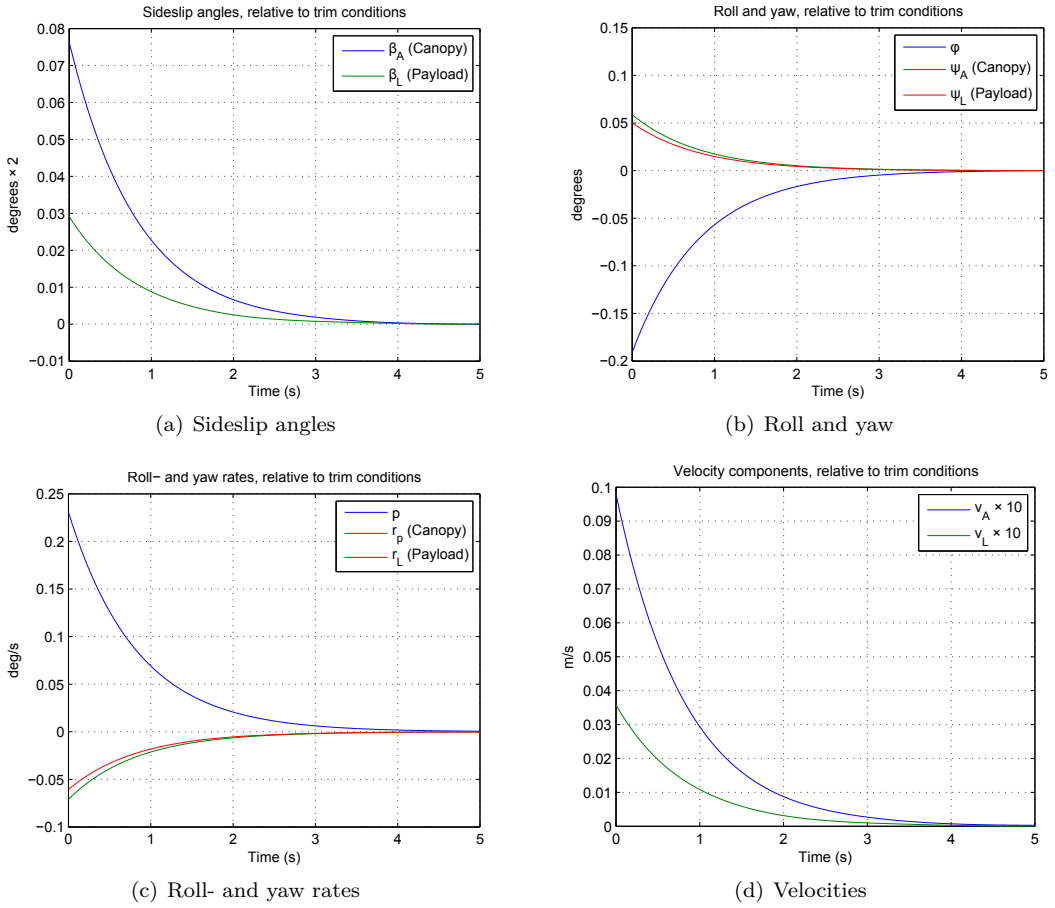
(c) Roll- and yaw rates



(d) Velocities

Figure 3.10 – Lateral real pole 1 (λ_{12}) response.

The modes behave slightly different to the real lateral modes expected in conventional aircraft, as also found by Hur [24]. Classically, two lateral real poles represent the roll subsidence and spiral modes for fixed wing aircraft. Roll subsidence is usually a relatively



(a) Sideslip angles

(b) Roll and yaw

(c) Roll- and yaw rates

(d) Velocities

Figure 3.11 – Lateral real pole 2 (λ_{15}) response.

fast mode which manifests as an exponential lag, or damping, in roll. The spiral mode is the tendency of the aircraft to return to (converge), or diverge from wings level flight following a roll disturbance. It is not uncommon for the spiral mode to be close to the stability margin and even unstable, referred to as a spiral divergence where the aircraft slowly diverges in roll, sideslip velocity, yaw and eventually altitude.

Real poles 1 and 2 show damped, converging responses, with λ_{15} mainly manifesting in roll and roll rate while λ_{12} displaying a combined roll and yaw response. The modes are stable and reasonably fast. Both are largely controllable by δ_a and wind input as shown below for real pole 1,

$$\left[\dot{z}_{12}\right] = \mathbf{F}_{m_{12}}\left[z_{12}\right] + \mathbf{G}_{m_{12}}\left[\delta_a\right] + \mathbf{W}_{m_{12}}\left[w_y\right] \tag{3.2.46}$$

$$\mathbf{G}_{m_{12}} = \left[-6.02\right] \qquad \mathbf{W}_{m_{12}} = \left[-69.47\right] \tag{3.2.47}$$

and for real pole 2:

$$\left[\dot{z}_{15}\right] = \mathbf{F}_{m_{15}}\left[z_{15}\right] + \mathbf{G}_{m_{15}}\left[\delta_a\right] + \mathbf{W}_{m_{15}}\left[w_y\right] \tag{3.2.48}$$

$$\mathbf{G}_{m_{15}} = \left[5.89\right] \qquad \mathbf{W}_{m_{15}} = \left[32.37\right] \tag{3.2.49}$$

3.3 Linear Analysis Conclusion

In this chapter a linear model was numerically computed from the non-linear MATLAB simulation. It was shown that the linear model compares well to the non-linear model for small perturbations, especially for the lateral motion. The linear model poles were then analysed, from which some insight was gained into the natural modes of motion of the parafoil.

Longitudinal complex pole pairs 2 and 3 represent the payload pendulum mode and short period mode, respectively. Both modes are very fast and die away in well under a second, imposing no adverse effects on the longitudinal motion of the parafoil. The phugoid motion is well damped compared to classical fixed wing aircraft, dying out after only 2 periods of oscillation. The effect of the longitudinal modes on the flight is acceptable and it is deemed unimportant to provide artificial damping.

The lateral dutch roll mode is well enough damped and fast enough to be ruled out as detrimental to the parafoil flight. Little will be gained from attempting to control this mode. Both lateral real poles are stable, reasonably fast and do not require additional attention.

Section 3.2.2.2 presented the payload relative twist mode which is very poorly damped and could result in persistent oscillations if it is not considered when designing the lateral control system. Taking care not to actuate at frequencies which could excite the mode will prevent the control system from inducing or worsening oscillations. This will be addressed in the next chapter, where the lateral model will be used to design the flight control system.

Chapter 4

Flight Control System Design

Flight control system (FCS) design is divided into the following sections:

- Stability augmentation
- Control
- Guidance
- Path planning and mission control

The structure is depicted graphically in Figure 4.1.

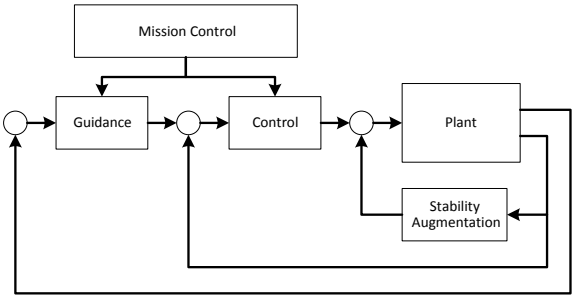


Figure 4.1 – Block diagram of FCS.

At the lowest level, or inner-most loop, stability augmentation damps or prevents the excitation of the unwanted natural modes of motion. Control implements feedback compensation, allowing the attitude of the parafoil to be controlled. By commanding this, the guidance controller ensures that an appropriate trajectory is flown in order to arrive at a desired position. Path planning and mission control overlooks guidance and control and is responsible for planning appropriate trajectories and performing top level decisions to ensure that the vehicle reaches its target location. This chapter will focus on stability augmentation, control and guidance, providing the platform which will be employed by the mission control logic to fulfil the flight objectives, described in Chapter 5.

Control and guidance design will consider only the lateral parafoil dynamics. The lateral dynamics can safely be considered separate from the longitudinal dynamics for small perturbations around the trim condition. Longitudinal effects induced by lateral motion such

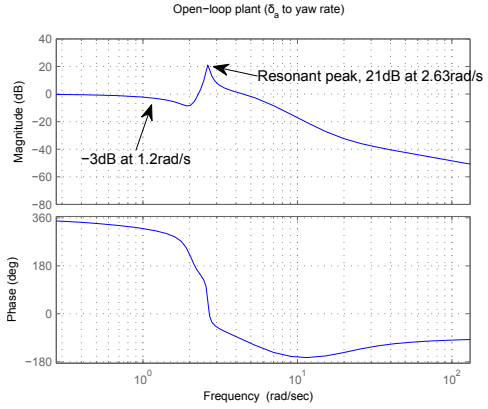


Figure 4.2 – Frequency response of open loop plant, δ_r input to $\dot{\psi}$ output.

as increased sink rate, will be accounted for by the mission control. The decision of considering only lateral control is further motivated by the fact that little longitudinal glide slope authority is available (with exception of the flare manoeuvre, which is outside the scope of this chapter) and that the longitudinal natural modes of motion are stable, do not adversely affect the flight and thus do not require stability augmentation.

4.1 Control System Overview

The lateral control authority available is asymmetric brake deflection (δ_a), which induces a yaw rate when deflected. It is thus natural for the inner-most control loop to regulate yaw rate. Stability augmentation will be considered for the payload relative twist oscillation in Section 4.2 after which a yaw rate controller will be designed in Section 4.3 to regulate yaw rate, establishing the inner control loop and enabling the subsequent controllers to guide the parafoil.

The lateral model can be written in state space form as

$$\dot{\mathbf{x}}_{lat} = \mathbf{A}_{lat}\mathbf{x}_{lat} + \mathbf{B}_{lat}\mathbf{u}_{lat} \tag{4.1.1}$$

$$\mathbf{y}_{lat} = \mathbf{C}_{lat}\mathbf{x}_{lat} \tag{4.1.2}$$

where $\mathbf{x}_{lat}$ is the lateral state vector, $\mathbf{A}_{lat}$ is the lateral system matrix, $\mathbf{B}_{lat}$ the input matrix, $\mathbf{u}_{lat}$ the input (δ_a), $\mathbf{C}_{lat}$ the output matrix and $\mathbf{y}_{lat}$ the model output. $\mathbf{C}_{lat}$ is chosen to extract yaw rate.

The servo drive system for the parafoil was designed to allow continuous sinusoidal steering actions of approximately 40% deflection to be performed at 1 Hz [17]. While the motors can perform better, this is a convenient specification to work from. The actuator response is approximated by a first-order low pass filter with a bandwidth of 2π:

$$H_\delta = \frac{2\pi}{s + 2\pi} \ . \tag{4.1.3}$$

The frequency response of the linear parafoil model for δ_r input to yaw rate output, depicts the resonant peak of the payload relative twist mode at 2.63 rad/s in Figure 4.2. The peak falls within the bandwidth of the actuators, making it possible to disturb the mode through control.

4.2 Yaw Rate Oscillation Damper

The payload relative twist oscillation can either be induced by external disturbances such as wind gusts, or by the control system itself. The former case is addressed by providing active damping when the oscillation arises, while the latter is addressed by preventing the control system from commanding actuation in the frequency band of the natural mode of motion. These two approaches are illustrated in Figure 4.3, which shows a block diagram of the inner lateral control loops. The *plant* block represents the lateral vehicle dynamics, in-

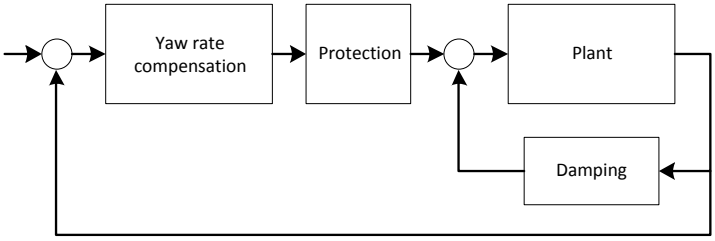


Figure 4.3 – Block diagram of inner lateral control loops.

cluding the actuator dynamics. The inner-most loop containing the *damping* block provides stability augmentation by damping out unwanted oscillations appearing at the plant output. The *protection* block prevents unwanted frequency components in the controller commands from reaching the plant. This configuration prevents the control system from exciting the unwanted mode of motion and damps out the effect of external disturbances.

This section considers the design of stability augmentation in order to actively damp the payload yaw rate oscillation when induced by external disturbances, represented by the *damping* block in Figure 4.3. The compensation will attempt to damp out unwanted yaw rate oscillations while not interfering with lower frequency yaw rate commands.

Yaw rate measurements are updated at 50 Hz. The fastest system dynamics are at 2.63 rad/s (0.37 Hz), as seen in the open loop plant bode plot in Figure 4.2. The sample frequency of the measurements is more than 35 times the frequency of the fastest dynamics which allows the yaw rate controller to be designed by emulation, i.e. in the continuous time domain [19].

As a first attempt to provide damping, a feedback loop is closed around the plant, feeding back yaw rate to the actuator input. Figure 4.4 shows the root locus plot for the yaw rate feedback. Due to the very high resonant peak, a small proportional feedback gain of

$$K_{damper} = 0.382 \tag{4.2.1}$$

is able to suppress lower frequencies in the feedback path while feeding back frequencies at the resonant frequency. The closed loop pole damping is increased from 0.02 to 0.05. However, apart from its limitations as a compensator, the proportional feedback loop may also reduce the overall system bandwidth. Since not only the unwanted frequencies are selectively damped, it will also attempt to counter yaw rate commands.

In order to investigate better damping, a filter is considered in the feedback path which will

- cut off frequencies below the desired yaw rate controller bandwidth so that this damper does not attempt to counter yaw rate commands, and
- provide appropriate phase at the resonant frequency of the oscillatory mode in order to enable maximum damping.

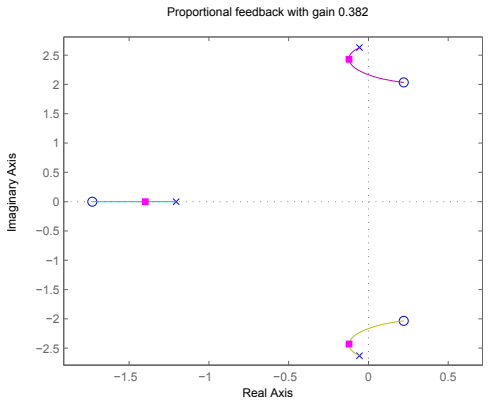


Figure 4.4 – Root locus plot of plant with proportional feedback.

The design is based on a sharp cut-off band pass filter, with the transfer function

$$D_{peak}(s) = \frac{s^2 + as + \omega_c^2}{s^2 + bs + \omega_c^2}, \; a > b \; . \tag{4.2.2}$$

Parameters a and b are directly related to the damping ratios of the complex conjugate zero and pole pair, respectively, and are used to tune the sharpness of the filter. The centre frequency, ω_c, is selected so that the phase change counters the phase induced by the plant in order to achieve maximum damping. An appropriate filter transfer function is

$$D_1(s) = 0.01 \times \frac{s^2 + 4.77s + (2.4)^2}{s^2 + 0.392s + (2.4)^2} \; . \tag{4.2.3}$$

The bode plot for this initial design is shown on the left-hand side of Figure 4.5. The filter cuts off high enough so that lower frequency yaw rate commands will not be affected and provides a negative phase to counter the phase of 48.2° at 2.63 rad/s shown in Figure 4.2.

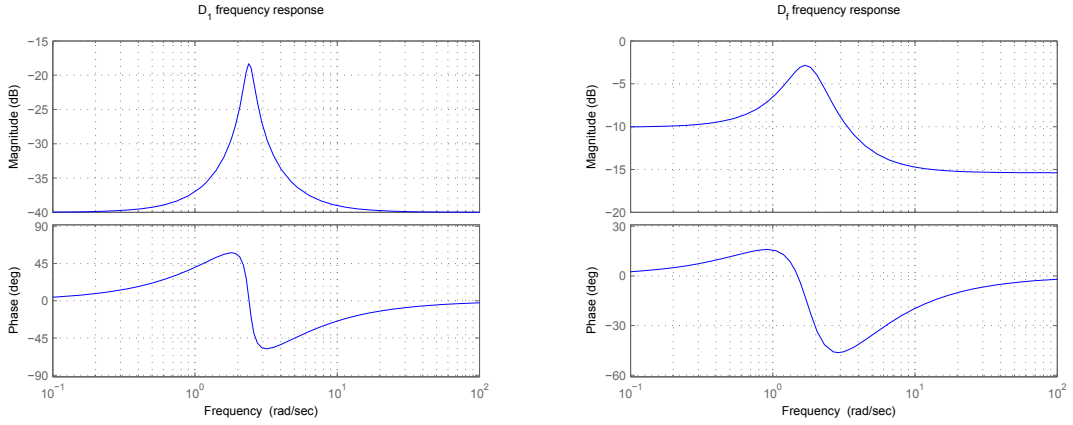


Figure 4.5 – Frequency response of band pass (peak) filter D_1 (left) and modified filter D_f (right).

As a final design step, the root locus for the plant combined with the filter is considered where the feedback gain and the position of the complex conjugate pole pair of the filter can be adjusted in order to fine tune the damping of the closed loop poles of the system.

Figure 4.6 shows the unchanged root locus of the combined system on the left-hand side and the final root locus where the poles have been adjusted, on the right-hand side. The closed loop pole damping for the final system on the right hand side is 0.2, a drastic improvement over the 0.02 of the open loop plant. The modified filter frequency response is given on the

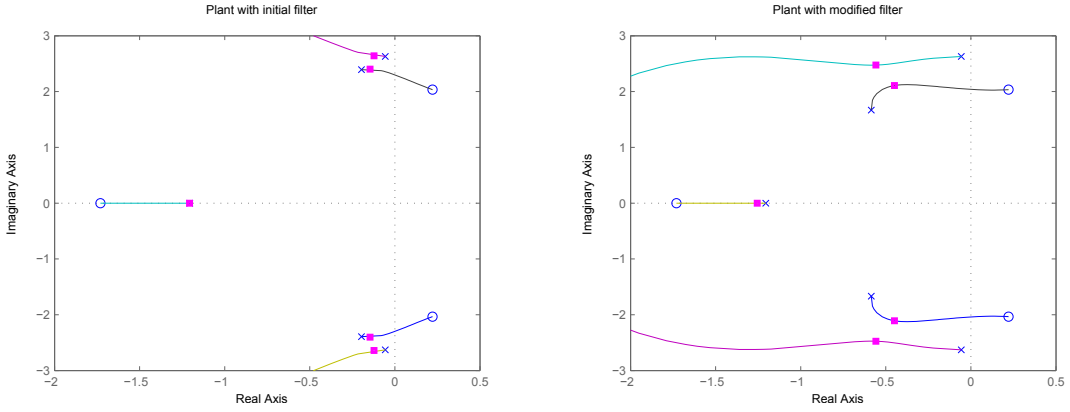


Figure 4.6 – Root locus plot of yaw rate damper with initial band pass filter D_1 (left) and filter with adjusted poles D_f (right).

right-hand side of Figure 4.5 and the transfer function is

$$D_f(s) = 0.17 \times \frac{s^2 + 4.7s + 5.76}{s^2 + 1.17s + 3.12} \ . \tag{4.2.4}$$

Figure 4.7 illustrates the damping effects of the proportional feedback and peak filter (D_f) compensation methods. The linear simulation was initialised with the real part of the payload twist mode eigenvector in order to induce the unwanted oscillation. Both compensation methods damp out the oscillation, with the filter method being most effective. The resulting actuator command (δ_a) due to D_f, depicted in Figure 4.8, experiences a maximum peak amplitude of roughly half the amplitude of the yaw rate oscillation. This allows reasonably large yaw rate oscillations to be countered without exceeding the maximum allowable deflection.

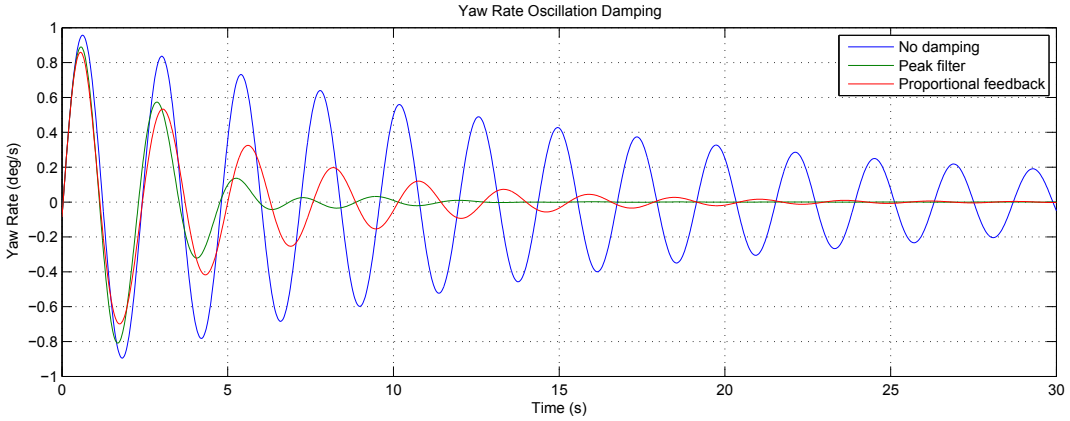


Figure 4.7 – Yaw rate damper linear simulation results for payload twist mode.

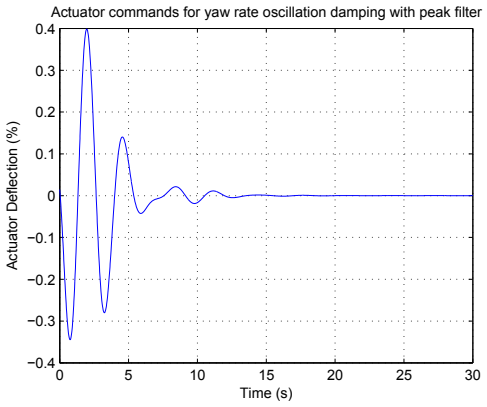


Figure 4.8 – δ_a command history of D_f damping the payload twist oscillation.

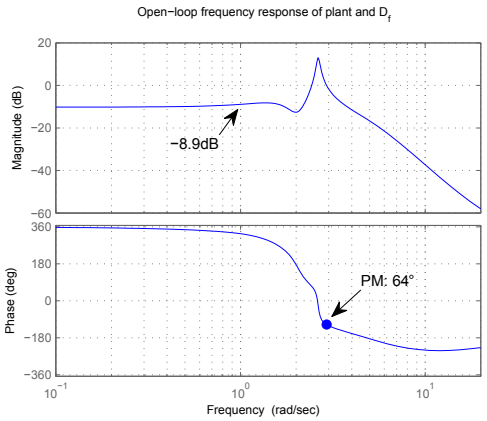


Figure 4.9 – Frequency response of open loop plant and D_f.

Figure 4.9 illustrates the frequency response of the open loop, cascaded damper and plant (with the actuator dynamics of Equation (4.1.3) included). The low frequency gain of -8.9 dB prevents the damper from reacting to low frequency yaw rate commands. The phase margin of 64.4° is also indicated which ensures some robustness against unmodeled actuator lag.

D_f can be expressed in state space form as

$$\dot{\mathbf{x}}_f = \mathbf{A}_f\mathbf{x}_f + \mathbf{B}_f\mathbf{u}_f \tag{4.2.5}$$
$$\mathbf{y}_f = \mathbf{C}_f\mathbf{x}_f \tag{4.2.6}$$

and appended to the lateral model in the feedback path as depicted in Figure 4.3, with the resulting system state space form given by

$$\begin{bmatrix}\dot{\mathbf{x}}_{lat}\\ \dot{\mathbf{x}}_f\end{bmatrix} = \begin{bmatrix}\mathbf{A}_{lat} & \mathbf{B}_{lat}\mathbf{C}_f\\ \mathbf{B}_f\mathbf{C}_{lat} & \mathbf{A}_f\end{bmatrix}\begin{bmatrix}\mathbf{x}_{lat}\\ \mathbf{x}_f\end{bmatrix} + \begin{bmatrix}\mathbf{B}_{lat}\\ 0\end{bmatrix}\mathbf{u} \tag{4.2.7}$$
$$\mathbf{y} = \begin{bmatrix}\mathbf{C}_{lat} & \mathbf{0}\end{bmatrix}\begin{bmatrix}\mathbf{x}_{lat}\\ \mathbf{x}_f\end{bmatrix} . \tag{4.2.8}$$

It is desirable for the closed loop system to have unity DC gain for yaw rate input (in °/s) to δ_a output. This corresponds to a gain of

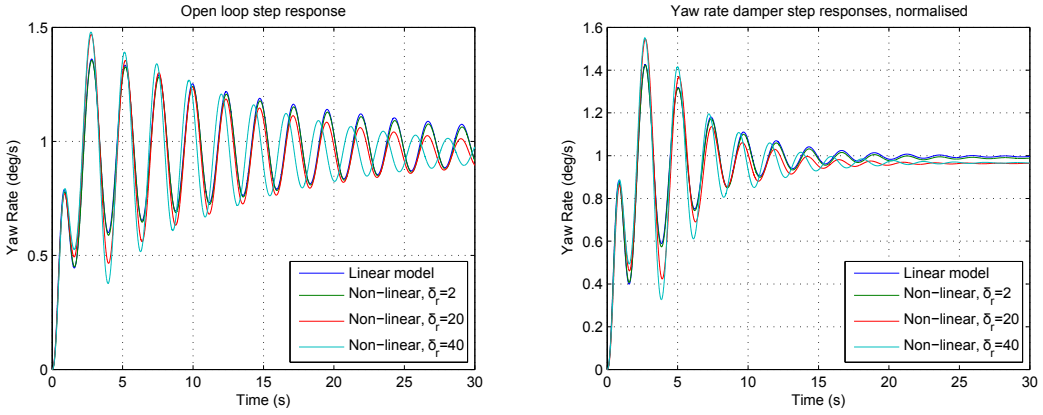
$$K_{\delta yrd} = 1.17 . \tag{4.2.9}$$

The yaw rate damper design presented in this section is applicable for the specific parafoil model used. If the damper were to be implemented on a parafoil with a payload twist oscillation mode at a different frequency, the design would have to be revised for the applicable model.

4.2.1 Non-linear Simulation

As a final test, the yaw rate damper is applied to the non-linear parafoil model for a yaw rate oscillation induced by a sudden δ_a step input. Firstly, Figure 4.10(a) gives several open loop step responses of different magnitudes, normalised, for the linear and non-linear models. The differences from the linear model due to the unmodeled non-linearities are evident in the larger brake deflections. Similar normalised step responses with the yaw rate damper

enabled are given in Figure 4.10(b). The damper performs well for all step responses, despite the differences from the linear case. The differences in the steady state values are due to the non-linear effects of larger brake deflections. The good responses verify the usefulness and applicability of the linear model as well as robustness of the damper for small model inaccuracies.



(a) Open loop step responses.

(b) Yaw rate damper step responses.

Figure 4.10 – Non-linear simulation step responses with and without the yaw rate damper.

4.3 Yaw Rate Controller

On the lowest level, the attitude of the parafoil is controlled by controlling the yaw rate with asymmetric brake deflection. Rate gyroscope measurements provide yaw rate in body axes (R) while the yaw rate controller is designed using yaw rate in inertial axes ($\dot{\psi}$). The relationship of inertial yaw rate to the body axes rates is given by

$$\dot{\psi} = Q \sin\phi \sec\theta + R \cos\phi \sec\theta, \quad |\theta| \neq \frac{\pi}{2} \tag{4.3.1}$$

and for small roll (ϕ) and pitch (θ) angles,

$$\dot{\psi} \approx R \tag{4.3.2}$$

Since the steady state roll and pitch angles change with an increase in δ_a, as seen in Figure 2.7, Equation (4.3.2) will only hold for small deflections. Figure 4.11 illustrates the divergence of $\dot{\psi}$ and R for increasing brake deflections. If gyroscope measurements, R, are to be used for the yaw rate controller, the heading controller performance will be affected for large heading commands that require a brake deflection of more than 20%. However, this effect is minimal and it is thus deemed acceptable to use either R or $\dot{\psi}$ as measurements for the yaw rate controller.

As mentioned in the previous section, yaw rate measurements are updated at 50 Hz, 35 times the frequency of the fastest plant dynamics, allowing the yaw rate controller to be designed by emulation, i.e. in the continuous time domain [19].

Figure 4.12 depicts a block diagram of the yaw rate controller design. The plant refers to the linear lateral model described by Equations (4.1.1) and (4.1.2) with the actuator dynamics included. It is required that the input and output units are angular rate in degrees per second. The output matrix $\mathbf{C}_{lat}$ extracts yaw rate, and is scaled to convert from radians

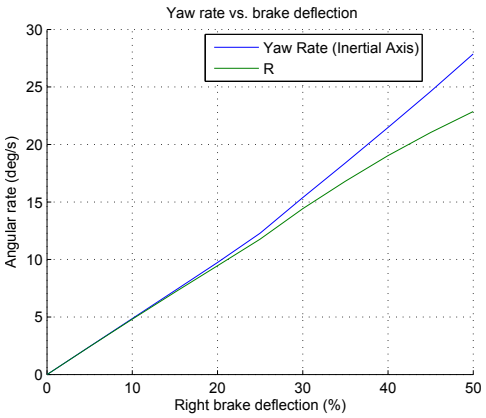


Figure 4.11 – $\dot{\psi}$ and R vs. asymmetric brake deflections.

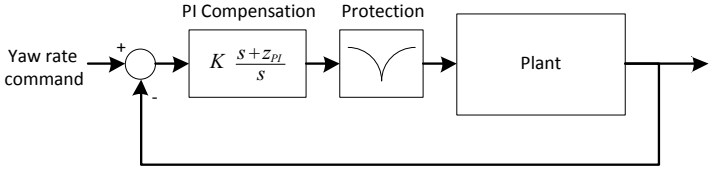


Figure 4.12 – Yaw rate controller block diagram.

to degrees. The input matrix $\mathbf{B}_{lat}$ is scaled with a factor K_δ which relates turn rate in degrees per second to asynchronous brake deflection, δ_a, so that the plant transfer function has unity DC gain,

$$K_\delta = 2\ . \tag{4.3.3}$$

When implementing the control on a specific parafoil, K_δ can be determined experimentally for that parafoil and updated in the control system. For instance, a constant open loop asymmetric brake deflection, δ_a, can be applied and the resulting steady state yaw rate can be recorded. K_δ can then be solved from

$$\delta_a = K_\delta \dot{\psi}\ . \tag{4.3.4}$$

The first task in designing the yaw rate controller is to provide some form of "protection" to the plant from the yaw rate commands against inducing the unwanted dynamics. Since the model displays a very distinct resonant peak, as evident in the frequency response in Figure 4.2, a notch filter is ideal for preventing frequency components in the specific band to reach the plant. A common transfer function for realising a notch filter is

$$D_{notch}(s) = \frac{s^2 + 2\zeta_z\omega_o s + \omega_o^2}{s^2 + 2\zeta_p\omega_o s + \omega_o^2},\ \zeta_p > \zeta_z \tag{4.3.5}$$

where ω_o is the centre frequency of the notch and ζ_z and ζ_p are the damping ratios of the complex conjugate zero pair and pole pair, respectively. ω_o is chosen as 2.63 rad/s, the resonant frequency of the unwanted mode of motion. ζ_z and ζ_p are then chosen such that the filter notch is deep enough and cuts off sharp enough to allow adequate bandwidth for the yaw rate commands. Seen from another perspective, the zeros of the filter can be placed to cancel the unwanted poles. The filter poles are then placed so that the filter shape is

adequate as mentioned above. The filter is depicted in Equation (4.3.6) and the filter bode plot is shown in Figure 4.13. Note the cut-off frequency around 1 rad/s and the depth of approximately -25 dB for which the filter was tuned. Figure 4.14 illustrates how the notch filter prevents the payload twist oscillation when a step input is applied to the open loop plant.

$$D_{notch}(s) = \frac{s^2 + 0.254s + 6.933}{s^2 + 4.54s + 6.949} \tag{4.3.6}$$

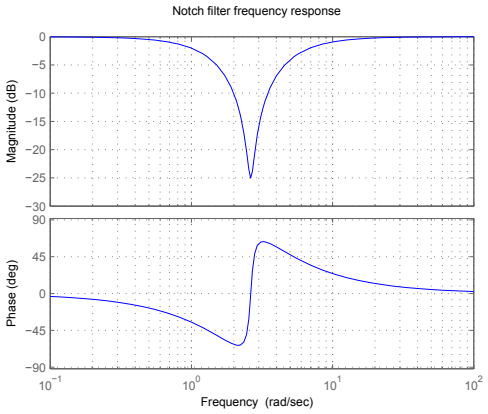


Figure 4.13 – Frequency response of notch filter D_{notch}.

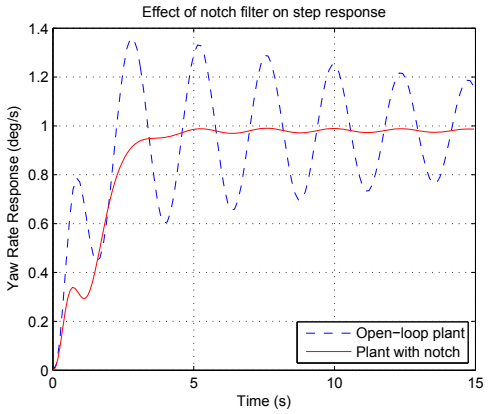


Figure 4.14 – Open loop step response with and without notch filter.

With the notch compensation preventing the excitation of the unwanted dynamics, focus is shifted to controlling yaw rate. Due to the fact that the relationship in Equation (4.3.4) varies slightly for different brake deflections, an integrator is required to ensure zero steady state error. Proportional and integral (PI) control is implemented to provide a fast response time. The PI compensation is considered in the form of

$$D_{PI}(s) = K_{\dot{\psi}} \frac{s + z_{PI}}{s} \ . \tag{4.3.7}$$

The zero location, z_{PI}, and gain $K_{\dot{\psi}}$ are tuned for an appropriate response. The following values give the desired response:

$$K_{\dot{\psi}} = 1, \qquad z_{PI} = 0.83 \ . \tag{4.3.8}$$

The resulting yaw rate controller transfer function is thus

$$D_{\dot{\psi}}(s) = \frac{(s + 0.83)(s^2 + 0.254s + 6.933)}{s(s^2 + 4.54s + 6.949)} \ . \tag{4.3.9}$$

The closed loop step response of the plant with the PI and notch compensation is plotted in Figure 4.15. The plot illustrates a reasonably fast response with an overshoot of less than 20% and a rise time of less than 3 s.

The notch and PI configuration developed above was implemented in a flight test on 25 April 2013, as documented in Section 8.2. Figure 8.2 shows the yaw rate response to a yaw rate controller step input. The parafoil responds to the step command after which the yaw rate controller attempts to counter an oscillation, but in doing so further excites the oscillation. The oscillation frequency is determined to be approximately 4.8 rad/s. Clearly

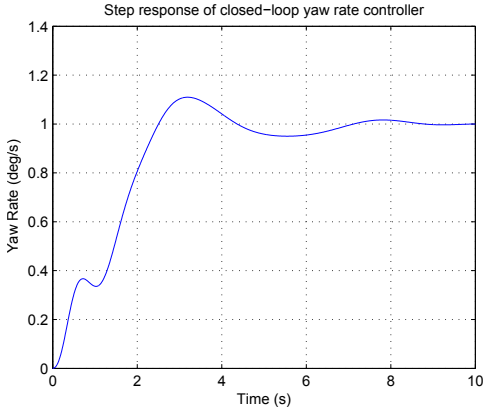


Figure 4.15 – Closed loop step response of the yaw rate controller with PI and notch compensation.

a discrepancy exists between the physical parafoil and the linear model and a serious design flaw in the yaw rate controller is exposed. While the notch filter attenuates frequencies in the range of the expected oscillation mode, it neglects to suppress higher frequencies, thereby exposing the control system to potential higher frequency dynamics.

The compensator has to be modified to eliminate the design flaw. If the exact frequency of the problematic oscillation is reliably known, the notch filter can be moved or an additional notch can be added to the compensation. This can be done with little effort but requires fine-tuning of the PI controller with the new notch filter. However, since the oscillation resulted from unanticipated dynamics in the first place, due to a lack of knowledge of the physical system, a more robust solution is to attempt to reject all higher frequencies in order to compensate for all unknown cases. This will not only alleviate the current problem, but also suppress possible other model discrepancies.

By shifting the complex pole pair of the notch filter in Equation (4.3.6) to a slight lower frequency, some attenuation of higher frequencies can be achieved while still maintaining the desired notch shape, as illustrated in Figure 4.16. Adding a real pole to serve as a low pass filter (LPF) further increases the high frequency attenuation. While the LPF alone is not adequate, since it reduces the available bandwidth (even with higher order LPFs), a simple first order LPF in combination with the notch filter provides the needed attenuation while not reducing the bandwidth, as shown in Figure 4.16.

The updated notch filter transfer function is given by

$$D_{notch2}(s) = \frac{s^2 + 0.254s + 6.933}{s^2 + 2.58s + 4.13} \ , \tag{4.3.10}$$

and the LPF pole location and newly tuned PI compensation transfer function are

$$p_{LPF} = -2 \ , \qquad D_{PI2}(s) = 0.677\frac{(s+0.9)}{s} \tag{4.3.11}$$

resulting in the complete compensator as

$$D_{\dot{\psi}2} = \frac{0.677(s+0.9)(s^2+0.254s+6.933)}{s(s+2)(s^2+2.58s+4.13)} \ . \tag{4.3.12}$$

The linear simulation results for a unity step input for the updated compensation is shown in Figure 4.17.

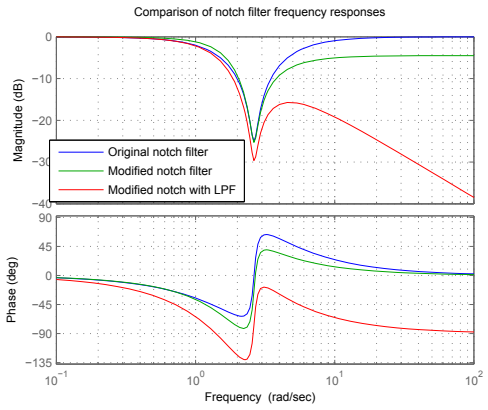


Figure 4.16 – Original and modified notch filter frequency responses.

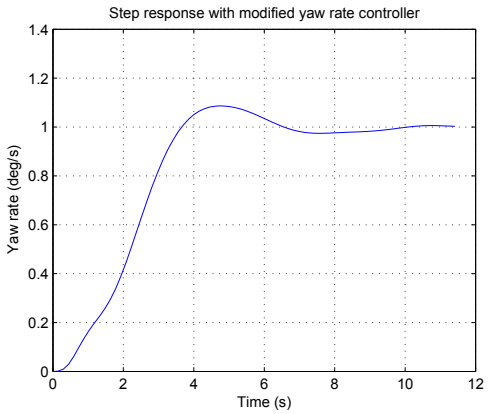


Figure 4.17 – Linear simulation step response of $D_{\dot{\psi}2}$.

If $D_{\dot{\psi}2}$ is represented in state space form as

$$\dot{\mathbf{x}}_{yr} = \mathbf{A}_{yr}\mathbf{x}_{yr} + \mathbf{B}_{yr}\mathbf{u}_{yr} \tag{4.3.13}$$

$$\mathbf{y}_{yr} = \mathbf{C}_{yr}\mathbf{x}_{yr} \ , \tag{4.3.14}$$

the linear plant is augmented as follows:

$$\begin{bmatrix}\dot{\mathbf{x}}_{lat}\\ \dot{\mathbf{x}}_{yr}\end{bmatrix} = \begin{bmatrix}\mathbf{A}_{lat} & \mathbf{B}_{lat}\mathbf{C}_{yr}\\ -\mathbf{B}_{yr}\mathbf{C}_{lat} & \mathbf{A}_{yr}\end{bmatrix}\begin{bmatrix}\mathbf{x}_{lat}\\ \mathbf{x}_{yr}\end{bmatrix} + \begin{bmatrix}0\\ \mathbf{B}_{yr}\end{bmatrix}\mathbf{u} \tag{4.3.15}$$

$$\mathbf{y} = \begin{bmatrix}\mathbf{C}_{lat} & \mathbf{0}\end{bmatrix}\begin{bmatrix}\mathbf{x}_{lat}\\ \mathbf{x}_{yr}\end{bmatrix} \ . \tag{4.3.16}$$

In conclusion, it is shown that a notch filter can be used to reject the resonant mode of motion in the ideal case where the model matches the actual parafoil. However, due to discrepancies between the model and actual parafoil an additional LPF is necessary to attenuate higher frequencies.

4.3.1 Non-linear Simulation

As a final test, the yaw rate controller is evaluated with the non-linear simulation. Figure 4.18 compares the linear and non-linear simulation results for 1°/s and normalised 10°/s and 20°/s step responses of the yaw rate controller. For the 1°/s step command, the linear response matches the non-linear response. However, for the 10°/s and 20°/s commands, the output of the non-linear model is increasingly different from the linear case since the linear model does not take the large deviation from trim conditions into account. Nevertheless, the performance is deemed acceptable. The output is within the required bounds and the 20°/s step is a worst case test which generally would not be required in practice.

4.4 Heading Controller

The heading controller will use the velocity vector heading as the heading measurements. The inertial sate estimator provides a heading measurement at a rate of 50 Hz.

To obtain a desired heading, the heading error is fed to the yaw rate controller with a gain, K_h. The current yaw angle measurement is obtained by integrating the yaw rate controller

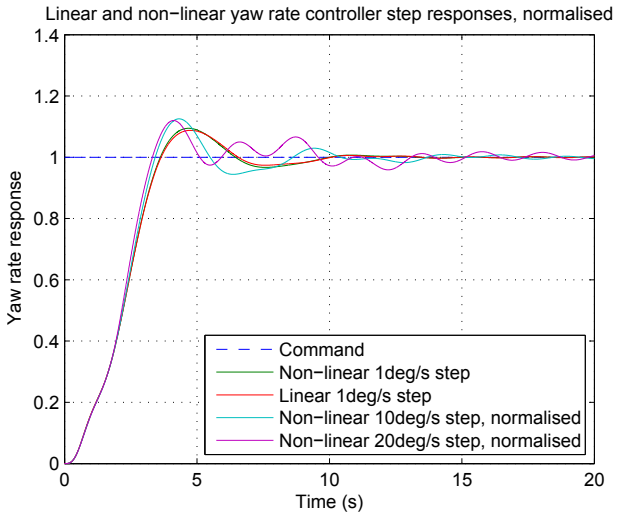


Figure 4.18 – Comparison of linear and non-linear simulation results for yaw rate controller step responses.

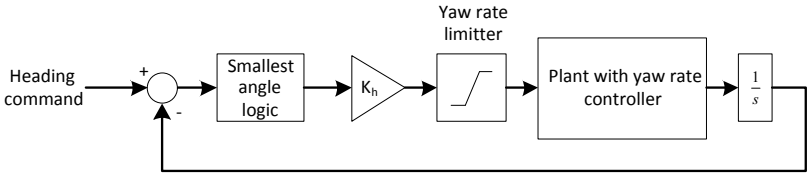


Figure 4.19 – Heading controller block diagram.

output. Saturation logic is also included to limit the input to the yaw rate controller for large yaw angle commands. The controller block diagram is shown in Figure 4.19.

The saturation limits are chosen based on the allowable actuator deflection. The saturation logic can be implemented to either limit the yaw rate command or the brake deflection command. Limiting the yaw rate command is chosen as it also has the effect of limiting the brake deflection while still allowing control authority for the yaw rate controller in the case where disturbances need to be countered. Increasing the saturation limit will allow the system to turn faster for large yaw commands, but might cause unwanted side-effects due to large asynchronous brake deflection. One such side-effect is the increased sink rate which was illustrated in Figure 2.7. The saturation limit, $\dot{\psi}_{lim}$, is chosen as

$$\dot{\psi}_{lim} = 20^\circ/\text{s} \tag{4.4.1}$$

so that

$$-\dot{\psi}_{lim} \leq \dot{\psi}_{cmd} \leq \dot{\psi}_{lim} \tag{4.4.2}$$

where $\dot{\psi}_{cmd}$ is the yaw rate command entering the yaw rate controller.

The primary restriction for determining an appropriate gain for the heading controller is a desire for minimal overshoot and a gain of

$$K_h = 0.23 \tag{4.4.3}$$

gives satisfactory results. The step response and root locus plot for this gain is shown in Figure 4.20.

The linear plant is augmented to include the heading controller dynamics as follows,

$$\begin{bmatrix} \dot{\mathbf{x}} \\ \dot{x}_h \end{bmatrix} = \begin{bmatrix} \mathbf{A} & -K_h\mathbf{B} \\ \mathbf{C} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ x_h \end{bmatrix} + \begin{bmatrix} K_h\mathbf{B} \\ 0 \end{bmatrix} \mathbf{u} \tag{4.4.4}$$

$$y = x_h \ , \tag{4.4.5}$$

where $\mathbf{x}$, $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ are, respectively, the state vector, system matrix, input matrix and output matrix of the linear plant including the yaw rate controller resulting from Equations (4.3.15) and (4.3.16), and x_h is the integrator state added by the heading controller.

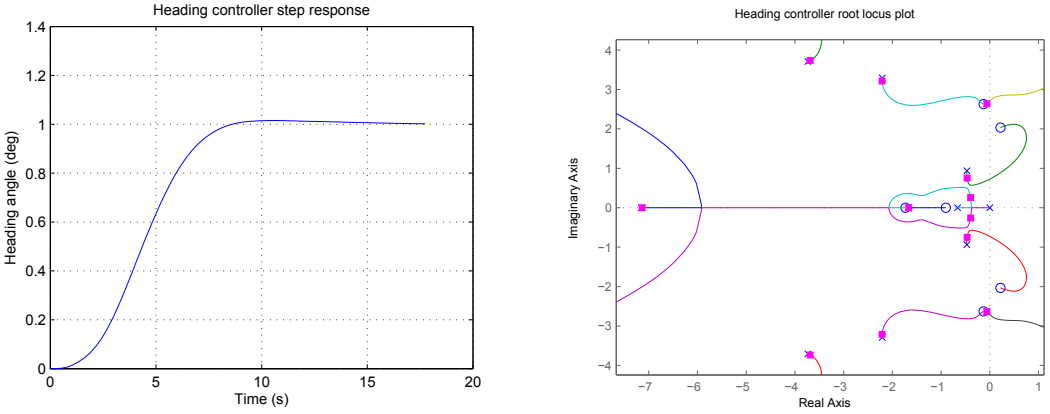


Figure 4.20 – Linear simulation step response (left) and root locus plot (right) for heading controller.

Finally, some form of logic should be included to choose the smallest angle when calculating the difference between the commanded and actual heading. For instance, if the current heading is 160° and the commanded heading is −160°, the issued command should be 40° clockwise in stead of 320° counter-clockwise. This logic is simple to implement and is represented by the *smallest angle logic* block in Figure 4.19.

Simulations with the non-linear model confirm that the controller behaves similar for the linear and non-linear case.

4.5 Path Follower

In a typical parafoil mission a trajectory is planned preflight based on the available altitude. A path follower is required for tracking the planned trajectory, with the desired specifications being minimal overshoot and a reasonably fast response.

Two path follow methods are considered in this project:

- Cross-track error guidance
- Non-linear guidance method by Sanghyuk Park [25]

Both methods will be discussed and evaluated next after which a suitable one will be chosen for implementation.

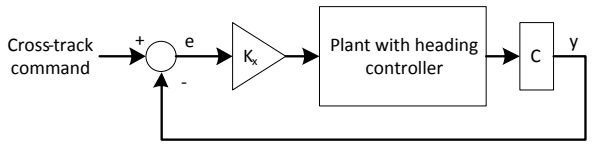


Figure 4.21 – Block diagram of cross-track error controller structure.

4.5.1 Cross-track Error Guidance

The cross-track error guidance controller feeds a cross-track error command to the heading controller, as depicted in the block diagram in Figure 4.21. The cross-track error, e, is the distance from the vehicle perpendicular to the path being followed and is illustrated in Figure 4.22. The sign of the error depends on the side of the path the vehicle is travelling on in the forward direction so that the resulting heading command will steer the vehicle closer to the path.

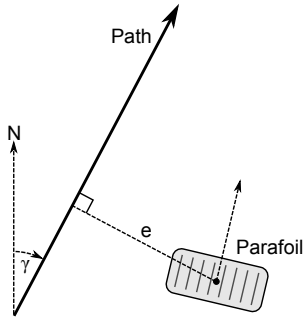


Figure 4.22 – Cross-track error.

To simplify controller design, consider the case where the desired path is on the x axis in the north direction (path angle $\gamma = 0$) and that the vehicle is flying parallel to the path, also in the north direction. Thus, the cross-track error in this case is the east (y) position of the vehicle. The output matrix, $\mathbf{C}$, is set to extract y. As with the heading controller, for proportional feedback control the restricting factor in determining the gain, K_x, is the minimum overshoot requirement. A gain of

$$K_x = 0.43 \tag{4.5.1}$$

satisfies the requirement at the trim forward velocity of 12.99 m/s, as illustrated in the unity step response and root locus plot in Figure 4.23. Simulations confirm that the performance with the non-linear model is similar.

The yaw rate controller relies on rate gyroscopes for yaw rate measurements which are likely to possess a bias of up to $\pm 0.5°/s$. This will result in a constant steady state cross-track offset as shown in Figure 4.24. For a bias of $0.5°/s$, the steady state cross-track error is approximately 5 m. The linear feedback control configuration allows for the addition of an integrator, which would ensure a zero steady state error, even in the presence of rate gyro errors. However, this will not be implemented in this project.

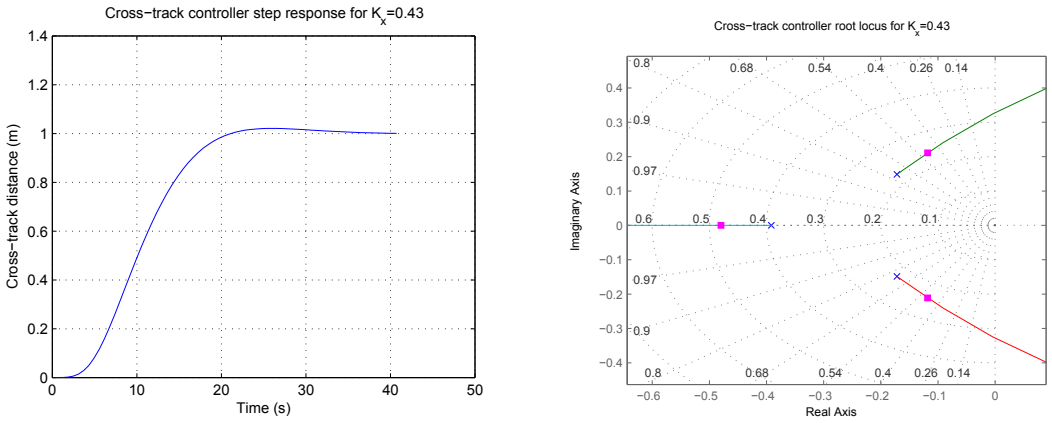


Figure 4.23 – Linear simulation step response (left) and root locus plot (right) for cross-track error guidance controller.

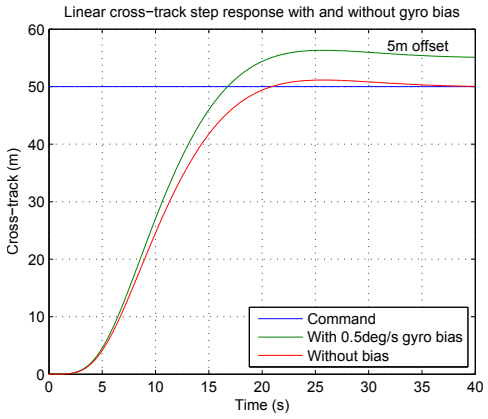


Figure 4.24 – Effect of 0.5°/s rate gyroscope bias on cross-track step input response.

The plant including the yaw rate controller and heading controller dynamics from Equations (4.4.4) and (4.4.5), is augmented with the cross-track error controller dynamics as follows:

$$\dot{\mathbf{x}} = \left[\mathbf{A} - K_x\mathbf{BC}\right]\mathbf{x} + \left[K_x\mathbf{B}\right]u \tag{4.5.2}$$

$$y = \mathbf{Cx}\ . \tag{4.5.3}$$

For practical implementation, the following points have to be taken into consideration:

- The cross-track error calculation has been reduced to simply the east location of the vehicle relative to the path for design purposes. In practice, the calculations are not so straight forward since the path angle γ and the parafoil location can be arbitrary.
- If the path is at some non-zero angle, the angle must be fed forward to the heading controller.
- When following a circle path segment that requires a constant yaw rate, a yaw rate feed forward should be applied to the yaw rate controller in addition to the heading feed forward:

$$\dot{\psi}_{ffd} = \frac{V_h}{R} \quad [\text{rad/s}] \tag{4.5.4}$$

- When transitioning from a straight line path segment to a circle, a sudden yaw rate feed forward command (Equation (4.5.4)) is issued at the start of the circle. The slow response of the yaw rate controller to a step command causes the parafoil to fly off-track by as much as 15 m for a circle radius of 100 m. Ramping up the yaw rate feed forward command from 0 to the desired value, starting some distance L_{ffd} before the start of the circle, gives the controller a chance to respond. A distance of

$$L_{ffd} = 50 \text{ m} \tag{4.5.5}$$

is found to work well, reducing the error to about 4 m. Figure 4.25 illustrates this improvement.

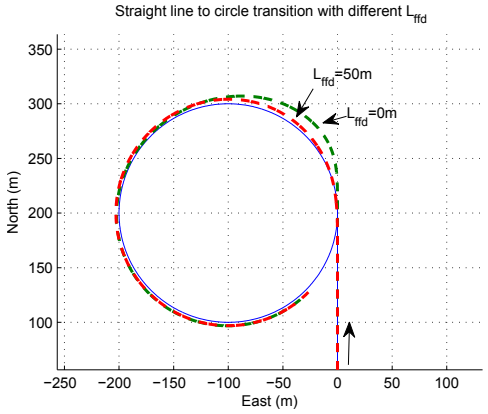


Figure 4.25 – Cross-track controller line to circle segment transition with different values of L_{ffd}.

4.5.2 Non-linear Guidance Method

Sanghyuk Park presents a non-linear lateral guidance method in [25] which is designed to tightly track a desired path with any curvature.

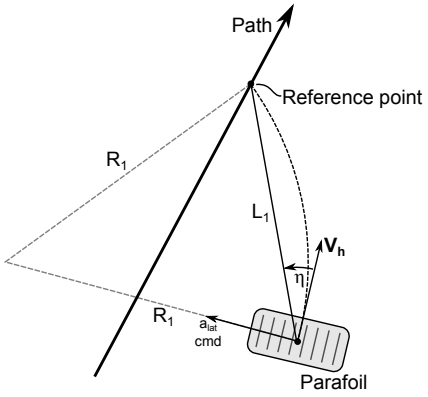


Figure 4.26 – Selection of the reference point for the non-linear guidance method.

At every instance of computation, the method starts by selecting a reference point on the desired path, in the forward direction, at a distance L_1 away from the vehicle, as illustrated in Figure 4.26. The angle between the current horizontal velocity vector $\mathbf{V}_h$ and the line connecting the vehicle and the reference point is denoted η. A lateral acceleration command of

$$a_{lat} = 2\frac{V_h^2}{L_1}\sin\eta \tag{4.5.6}$$

is then generated which is equal to the centripetal acceleration required to follow a circular path tangential to the vehicle velocity vector, intersecting the vehicle position and the reference point, as indicated by the dashed circle segment in the figure. The radius of this circle is

$$R_1 = \frac{L_1}{2\sin\eta}\ . \tag{4.5.7}$$

Thus, for the parafoil to follow this circle, a turn rate command of

$$\dot{\psi}_c = \frac{V_h}{R_1} = \frac{2V_h\sin\eta}{L_1}\ \ [\text{rad/s}] \tag{4.5.8}$$

must be issued. The sign of η depends on the direction of the line to the reference point relative to the velocity vector, and will result in a yaw rate command attempting to align the vehicle velocity vector to point to the reference point [25].

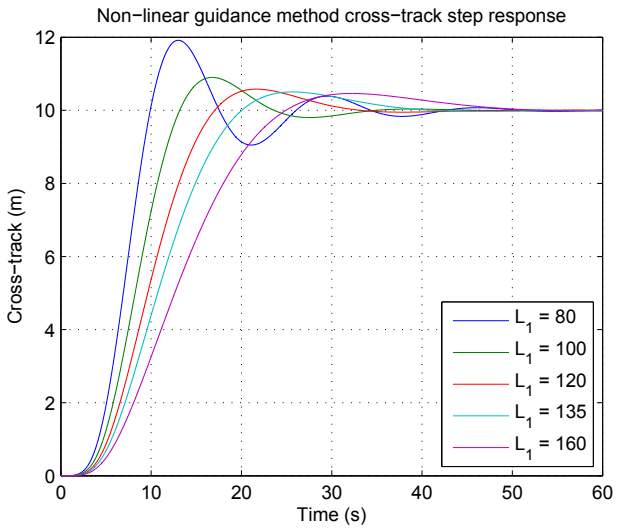


Figure 4.27 – Non-linear guidance controller step response for a 10 m cross-track step, with different values of L_1.

The non-linear guidance method is tuned by varying L_1, where a small value relates to a higher control gain and faster path convergence. Figure 4.27 shows the responses to a 10 m cross-track step for different values of L_1. Lower values display a shorter rise time and more overshoot. $L_1 = 135$ m results in a 100% rise time of 20 s, similar to the cross-track controller (Figure 4.23), yet with 5% overshoot as opposed to 2% for the cross-track controller. Varying L_1 between 120 m and 160 m affects the rise time but does not notably change the overshoot, however.

In the case of rate gyroscope bias, a steady state trajectory following error is experienced similar to that of the cross-track error controller shown in Figure 4.24.

The choice of L_1 dictates the minimum allowed turning radius R, since it is required that $L_1 < 2R$ for the method to be valid. In addition, the smaller L_1 and the change in path curvature, the smaller the resulting error in following the trajectory. One such case arises when transitioning from a straight flight path segment to a turning circle and is illustrated in Figure 4.28 where three transitions with different values for R are compared. Choosing R is a trade-off between the trajectory following error and the size of the turning circle. A value of

$$L_1 = 120 \text{ m} \tag{4.5.9}$$

is chosen which ensures an adequate rise time and overshoot response. Simulations with the non-linear parafoil model match those with the linear model extremely well.

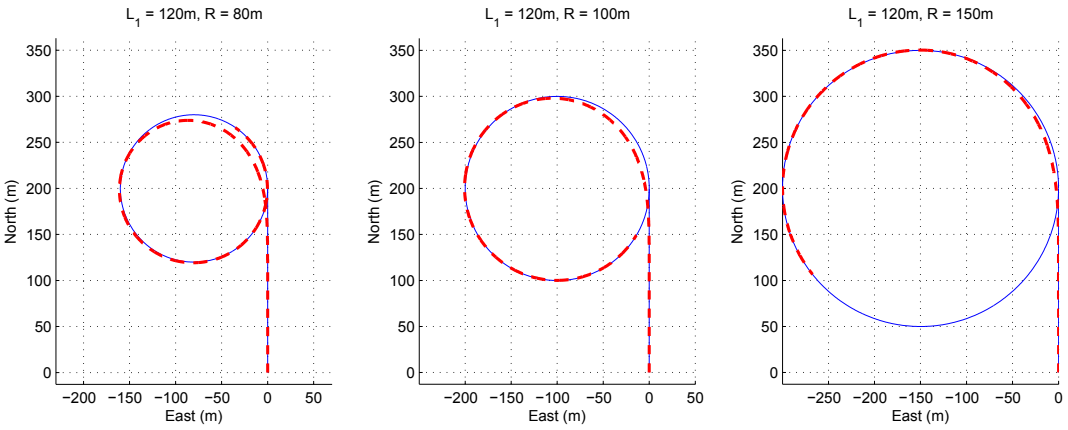


Figure 4.28 – Non-linear guidance controller line to circle segment transition.

4.5.3 Choosing A Path Follow Method

The non-linear guidance method uses future information about the path curvature and thus inherently compensates for sudden curvature changes (such as straight line to circular segment transition), constant circular paths and any other curvature. Transition between path segments does not cause sudden yaw rate command steps, but happens smoothly. The non-linear guidance method can also follow the terminal guidance trajectories (discussed in the next chapter) without modification and tweaks to the controller. Additionally, the non-linear guidance method takes the current vehicle velocity into account, as opposed to the cross-track error controller which is designed for a specific trim velocity.

The cross-track error controller structure allows for the addition of an integrator, which will ensure zero steady state error in the presence of rate gyroscope biasses. However, the steady state errors expected for the bias characteristics of the gyroscopes are acceptable for this project.

In the light of this, the non-linear guidance method is chosen for path following.

4.6 FCS Summary

In this chapter, different controllers were designed and combined, each building on the last by successively closing loops to form a FCS capable of tracking a trajectory while attempting not to excite undesired natural modes of motion.

First, the yaw rate damper was introduced as a means to damp out the unwanted payload oscillatory twist mode of motion. This was achieved by implementing a filter in the feedback path of the plant. The tuning of the filter is very specific to the model of the plant, however, and will not be effective and even cause adverse effects when applied to a plant with different characteristics and mode frequencies. This can also be caused by unmodeled characteristics such as delays in actuators. Thus, for the damper to be applied with confidence, the parafoil, actuators and other system characteristics will have to be modeled more closely. The damper can be enabled and disabled without notably affecting the rest of the FCS, however.

The basis of the FCS is the yaw rate controller, which is realised with PI compensation and a modified notch filter to prevent the controller from exciting the unwanted plant dynamics.

Next, a heading controller was designed, closing a loop around the yaw rate controller. However, the heading controller is only used in combination with the cross-track error controller. Since the non-linear guidance method of [25] is chosen due to its more natural path following abilities, the heading controller will not be used as the non-linear guidance method directly commands a yaw rate.

With the parafoil able to follow a desired trajectory, the next tasks are to create a means of planning appropriate paths and to establish top level mission control logic in order to complete the flight objective.

Chapter 5

Path Planning and Mission Control

5.1 Introduction

With the guidance and control capabilities established in the previous chapter, focus shifts to path planning and mission control. The goal of path planning is to plan a path that will enable the vehicle to land as close to the landing target as possible, while mission control makes top level decisions based on the circumstances that arise during flight. As mentioned before, a typical flight contains the phases

1. Launch
2. Homing
3. Energy management (EM)
4. Terminal guidance (TG)
5. Final approach (FA)
6. Flare and landing

which are illustrated in Figure 5.1.

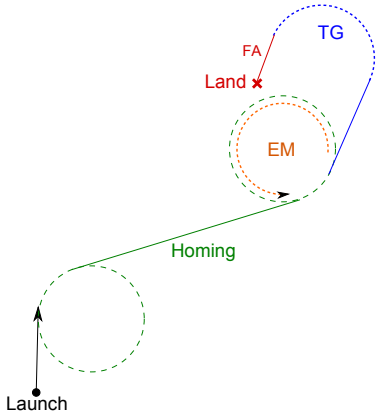


Figure 5.1 – Flight path overview

The path is constrained by the

- launch position and heading,
- minimum turn radius,
- landing target position and heading, and
- final approach distance.

In addition, the following variables are used in the planning:

- Vertical velocity, V_v
- Horizontal (forward) velocity, V_h

For preflight planning, the velocities are assumed to be constant for the entire flight. This is motivated by the minimum turn radius being large enough so that the maximum brake deflection used will not cause a significant change in velocity.

The TG section is the last chance to influence the accuracy of the landing and will thus be set up first, after which the rest of the flight is planned to bring the vehicle to the start point of the section. TG consists of a 180° turn leading into the FA phase as depicted in Figure 5.1 and will be discussed further in Section 5.2.

After the launch, the homing phase guides the vehicle through optional waypoints, if any, after which it will bring the vehicle to the start of the TG section. Simple Dubins paths[1] are planned for this, consisting of constant turns, connected by straight lines. If the height at which the homing section reaches the TG section is too large, an EM phase is added which will employ some strategy to eliminate the excess height. The homing and EM phases are discussed in Sections 5.3 and 5.4, respectively.

It is considered advantageous to choose the EM phase after the homing phase. Being in the region of the landing target, the area could be surveyed before landing and in the event that the flight time is drastically shortened due to unexpected circumstances, the chances of landing close to the target are increased. It has also been mentioned that the EM manoeuvres can be used for wind estimation. Wind estimation is especially useful for the TG phase of the flight and the most applicable when performed in the same region. In Section 5.4.1, a simple wind estimation technique, which only requires minimal modification to the EM phase, is discussed.

When planning the mission preflight, the altitude required by the path is calculated using velocity values which are expected for the vehicle. The ground station software, discussed in Section 6.4, provides an interface to the user for setting up the system with the desired mission objectives and will aid the user in choosing valid parameters based on the available flight time. After the launch, the flight path will be replanned using the measured altitude and velocities.

5.2 Terminal Guidance

The task of the terminal guidance phase is to ensure that the vehicle lands as close as possible to the landing target. This phase is the last chance to manoeuvre the vehicle and the terminal guidance scheme must compensate for initial position and other errors. In the worst case, the vehicle must land close to the landing target.

It is now assumed that the parafoil is brought to the start of the terminal guidance phase by the previous flight sections. Two methods of developing reference trajectories for terminal guidance are explored:

[1] A path consisting of an initial and final turning circle, connected with a tangential line.

- Modified Dubins trajectory
- Optimal method by Slegers and Yakimenko [8]

5.2.1 Modified Dubins Trajectory

Planning terminal guidance with a modified Dubins trajectory is a simple geometry based method. A path is planned at the start of the phase, after which it is followed until the final approach. The quality of this method can be enhanced by compensating for the effects of known wind and the vehicle characteristics when planning the path. Once planned, the accuracy of the method relies on the robustness of the path follower.

At the start of the terminal guidance phase, the parafoil is at some position relative to the target position, which is the starting point of the final approach. Ideally, the offset position (Δx, Δy, Δz) is such that the parafoil is facing in the opposite x-direction and a constant 180 ° turn can be performed to bring the parafoil to the target position in the available amount of time (or Δz). Thus,

$$\Delta x = 0, \tag{5.2.1}$$

$$\Delta y = L, \quad L \geq 2R_{nom}, \tag{5.2.2}$$

$$\frac{\Delta z}{V_v} = \frac{\pi \frac{L}{2}}{V_h}. \tag{5.2.3}$$

R_{nom} is a nominal turn radius which is larger than the minimum turn radius, R_{min}, by some factor in order to allow for path following corrections to be made during the turn. The left side of Equation (5.2.3) represents the time available given the height Δz, assuming a constant sink rate V_v and the right-hand side of the equation represents the time it will take to fly the constant circle assuming constant horizontal velocity V_h. If this equation holds, the starting position is at the ideal case as described above. This case is illustrated in Figure 5.2.

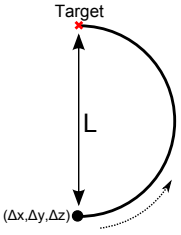


Figure 5.2 – Ideal case for initial position in terminal guidance phase.

If the constant turn time is less than the available time,

$$\frac{\pi \frac{L}{2}}{V_h} < \frac{\Delta z}{V_v}, \tag{5.2.4}$$

straight legs of equal length l are added to the start and end of the constant turn circle so that the time to travel the new path is equal to the available time:

$$\frac{\pi \frac{L}{2} + 2l}{V_h} = \frac{\Delta z}{V_v} \tag{5.2.5}$$

$$\Rightarrow l = \frac{\frac{\Delta z}{V_v} V_h - \pi \frac{L}{2}}{2}, \tag{5.2.6}$$

as depicted in Figure 5.3.

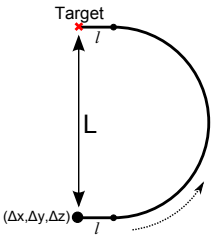


Figure 5.3 – Lengthened path where the constant turn time is less than the available time.

If the constant turn time is more than the available time,

$$\frac{\pi \frac{L}{2}}{V_h} > \frac{\Delta z}{V_v}, \tag{5.2.7}$$

the path is shortened by connecting the start and end point with a Dubins path; constant turning circles are fixed at the start and end point and connected with a straight line, as shown in Figure 5.4. The radius of the circles are equal and is adjusted so that the time spent travelling the path is equal to the available time:

$$\frac{(2(\frac{\pi}{2}R) + (L - 2R)}{V_h} = \frac{\Delta z}{V_v} \tag{5.2.8}$$

$$\Rightarrow R = \frac{\frac{\Delta z}{V_v} V_h - L}{(\pi - 2)}, \quad R_{nom} \leq R < \frac{L}{2} \tag{5.2.9}$$

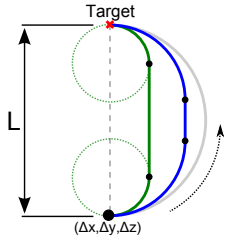


Figure 5.4 – Shortened path where the constant turn time is less than the available time.

5.2.2 Optimal Terminal Guidance

The Dubins trajectory method developed in the previous section has no compensation for constant and unknown changing winds that may affect the parafoil. Rather than attempting to estimate the constantly changing wind, [8] developed an optimal terminal guidance algorithm that accounts for the varying wind and other disturbances by adapting the trajectory during the final turn.

The algorithm calculates a TG trajectory as a solution to a two point boundary value problem (TPBVP) based on the inverse dynamics in the virtual domain [8]. The method assumes a constant known wind in the direction parallel to the FA and uses a simple 2D kinematic model to represent the vehicle movement and take the wind into account when planning the trajectory. Given a starting position and initial state, an optimal trajectory is generated for a specified time duration while satisfying the boundary values. The trajectory can be recalculated en route, allowing robustness to changing winds and tracking errors.

Per illustration, consider the simple constant turn rate turn with radius R, depicted in Figure 5.5, a typical result of the Dubins path planning described in the previous section. The trajectory satisfies the initial and final position boundary conditions and will be completed in a time duration of

$$T_{turn} = \frac{\pi R}{V_h} \ . \tag{5.2.10}$$

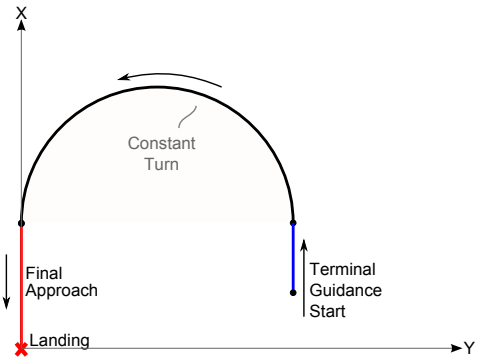


Figure 5.5 – Terminal guidance axis system.

For the same initial and final position constraints, the optimal algorithm can generate a trajectory which completes in exactly the same amount of time. Figures 5.6 and 5.7 compare the constant turn and the optimal trajectories for a turn with a radius of 100 m. Both trajectories satisfy the boundary conditions and both are completed in exactly the same time of 24.5 s. However, the optimal trajectory has the advantage of a smooth yaw rate and heading transition, as seen in Figure 5.7, since the algorithm takes the vehicle velocity and acceleration into account in addition to the position. Furthermore, the optimal algorithm can be recalculated at any point during the manoeuvre which allows for compensation for unknown wind errors.

The implementation of the algorithm with this project is discussed next. First, the task of setting up ideal boundary conditions is discussed, after which the algorithm for generating the optimal trajectory is given.

5.2.3 Setting Up Ideal Boundary Conditions for the Algorithm

Ideal boundary conditions for the optimal algorithm are set up based on a constant turn, depicted in Figure 5.5. This includes an ideal starting point (north, east, down and heading) for the TG phase which is used to plan the homing and EM phases, as well as a point where the TG turn will be initiated.

Note that the axis system origin for TG calculations is at the landing target.

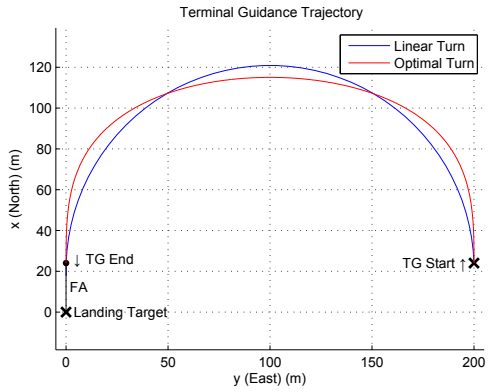


Figure 5.6 – Trajectory comparison of optimal terminal guidance turn and a linear turn with a radius of 100 m.

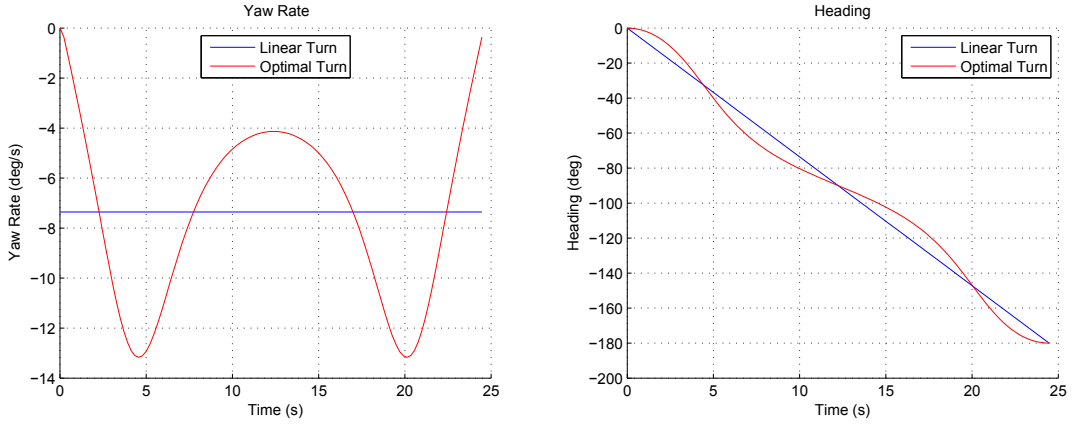


Figure 5.7 – Comparison of yaw rate and heading angle for an optimal terminal guidance turn and a linear turn with a radius of 100 m.

Firstly, the time span of the TG phase is determined, assuming that the horizontal velocity V_h and the vertical velocity V_v are constant.

T_{app}, the final approach time, is chosen to allow the parafoil to settle into a straight path before touchdown. The time can be chosen based on the height or time required to prepare for the flare manoeuvre. Generally, a longer time will allow for better correction of TG inaccuracy while a shorter time will allow less error to accumulate in the FA.

T_{turn}, the TG turn duration, is based on the time required for a constant turn, given by

$$T_{turn} = \frac{\pi R}{V_h} \tag{5.2.11}$$

where R is the radius of the constant turn and V_h is the horizontal velocity.

Yakimenko and Slegers [8] consider a setup as depicted in Figure 5.8 with a straight leg of length L, followed by the turn and finally the straight FA section. The turn start position is offset by a distance D_{switch} which is determined based on the wind velocity in order to ensure that the constant turn will end at the desired point. By representing the trajectory

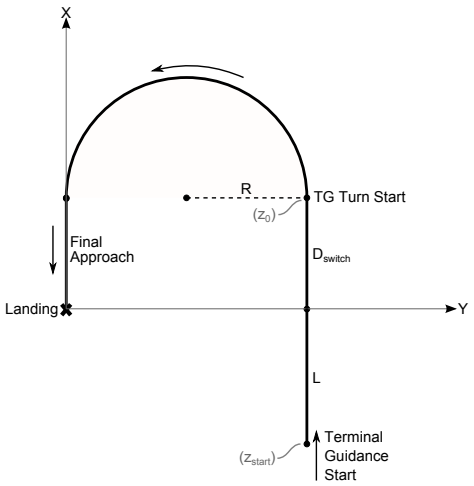


Figure 5.8 – Ideal TG setup.

as a simple kinematic model,

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} -W + V_h \cos\psi \\ V_h \sin\psi \\ V_v \end{bmatrix} \tag{5.2.12}$$

[8] integrates the inertial velocities along the x and y axes and derives equations for the starting position of the turn and the TG phase. The full derivation is available in [8]. Only the resulting equations are given below.

The altitude at which the TG phase starts, z_{start}, is given by

$$z_{start} = V_v \frac{L + V_h(T_{turn} + 2T_{app})}{W - V_h} \ . \tag{5.2.13}$$

The turn start x-coordinate, described by D_{switch}, is given by

$$D_{switch} = -\frac{z(V_h^2 - W^2) + V_h V_v T_{turn}(V_h - W) + x V_v (V_h + W)}{2V_h V_v} \ . \tag{5.2.14}$$

A wind velocity in the negative x-direction will result in a larger D_{switch}, while wind in the positive x-direction will cause D_{switch} to be smaller or even negative.

The sum of the FA and TG turn time, $(T_{app} + T_{turn})$, determines the altitude at the start of the TG turn as

$$z_0 = (T_{turn} + T_{app})V_v \ . \tag{5.2.15}$$

The boundary conditions for the ideal case are now set up. In a practice, however, the parafoil will not arrive at this position exactly and might have a non-ideal heading, yaw rate, velocity and acceleration. During flight, the TG turn start position will continually be calculated based on the actual vertical and horizontal velocities. The TG turn will be initiated when either

- the predetermined TG turn starting height z_0 has been reached, or
- the vehicle comes in line with the horizontal TG turn starting point (x-coordinate) defined by D_{switch},

whichever happens first. The optimal trajectory will then be planned with the current initial conditions, as discussed next.

5.2.4 Optimal Terminal Guidance Algorithm

Tables 5.1 to 5.3 summarise the boundary conditions, parameters and output variables of the optimal TG algorithm, respectively. In an ideal case, the initial conditions in Table 5.1 will correspond to the TG turn start point, described in the previous section. Otherwise, either the x_0 or z_0 state will match the ideal coordinates. Parameters V_h, V_v, T_{app} and T_{turn} are discussed in the previous section.

x_0, y_0, z_0	Initial position
ψ_0	Heading
$\dot{\psi}_0$	Yaw rate
$\dot{x}_0$, $\dot{y}_0$	Velocity
$\ddot{x}_0$, $\ddot{y}_0$	Acceleration
x_f, y_f	Final position
ψ_0	Final heading
$\dot{\psi}_0$	Final yaw rate
$\dot{x}_f$, $\dot{y}_f$	Final velocity
$\ddot{x}_f$, $\ddot{y}_f$	Final acceleration

Table 5.1 – Optimal terminal guidance algorithm boundary conditions

W	Wind velocity, positive in $-x$ direction (m/s)
V_h	Horizontal velocity (m/s)
V_v	Vertical velocity (positive downwards)
$\dot{\psi}_{max}$	Maximum allowable turn rate
T_{app}	Final approach time
T_{turn}	Terminal guidance turn time
N	Number of path nodes

Table 5.2 – Optimal terminal guidance algorithm parameters

	($i \in 1$ to N)
x_i, y_i, z_i	Position at each node
$\dot{x}_i$, $\dot{y}_i$	Velocity at each node
ψ_i	Heading at each node
$\dot{\psi}_i$	Turn rate command for each node
J	Cost function
t_i	Time span of each node

Table 5.3 – Optimal terminal guidance algorithm output

A basic overview of the algorithm is given here, with the necessary calculations to numerically generate the optimal trajectory. For an in-depth description, the reader may consult [8] and [26].

Yakimenko and Slegers [8] assume that the turn rate is low so that roll and sideslip angles can be ignored, and that the velocities are nearly constant which allows the horizontal trajectory kinematics to be represented by the following simple 2D kinematic model,

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} -W + V_h \cos\psi \\ V_h \sin\psi \end{bmatrix} . \tag{5.2.16}$$

[8] represents the solution of the TPBVP analytically as functions of a scaled abstract argument

$$\overline{\tau} = \tau/\tau_f \ \in [0;1] \ . \tag{5.2.17}$$

The boundary conditions, set up in terms of the model, are

$$\begin{bmatrix} x \\ y \end{bmatrix}_{\tau=0} = \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} \ , \ \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix}_{\tau=0} = \begin{bmatrix} -W + V_h \cos\psi_0 \\ V_h \sin\psi_0 \end{bmatrix} \ , \ \begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix}_{\tau=0} = \begin{bmatrix} -\dot{\psi}_0 V_h \sin\psi_0 \\ \dot{\psi}_0 V_h \cos\psi_0 \end{bmatrix} \tag{5.2.18}$$

and

$$\begin{bmatrix} x \\ y \end{bmatrix}_{\tau=\tau_f} = \begin{bmatrix} (V_h + W)T_{app} \\ 0 \end{bmatrix} \ , \ \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix}_{\tau=\tau_f} = \begin{bmatrix} -W + V_h \\ 0 \end{bmatrix} \ , \ \begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix}_{\tau=\tau_f} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \ . \tag{5.2.19}$$

The solution and its derivatives are assumed to be of the form

$$P_\eta(\overline{\tau}) = a_0^\eta + a_1^\eta \overline{\tau} + a_2^\eta \overline{\tau}^2 + a_3^\eta \overline{\tau}^3 + b_1^\eta \sin(\pi\overline{\tau}) + b_2^\eta \sin(2\pi\overline{\tau}) \ , \tag{5.2.20}$$

$$\tau_f P'_\eta(\overline{\tau}) = a_1^\eta + 2a_2^\eta \overline{\tau} + 3a_3^\eta \overline{\tau}^2 + \pi b_1^\eta \cos(\pi\overline{\tau}) + 2\pi b_2^\eta \cos(2\pi\overline{\tau}) \ , \tag{5.2.21}$$

$$\tau_f^2 P''_\eta(\overline{\tau}) = 2a_2^\eta + 6a_3^\eta \overline{\tau} - \pi^2 b_1^\eta \sin(\pi\overline{\tau}) - (2\pi)^2 b_2^\eta \sin(2\pi\overline{\tau}) \ , \tag{5.2.22}$$

where $(\eta = 1, 2)$, $(P_1(\overline{\tau}) = x(\overline{\tau}))$ and $(P_2(\overline{\tau}) = y(\overline{\tau}))$. The coefficients a_i^η and b_i^η are defined by the boundary conditions and resolve to

$$a_0^1 = x_0, \ a_1^1 = -(x_0 - x_f) - \frac{(2x_0'' + x_f'')\tau_f^2}{6}, \ a_2^1 = \frac{x_0''\tau_f^2}{2}, \ a_3^1 = -\frac{(x_0'' - x_f'')\tau_f^2}{6}, \tag{5.2.23}$$

$$b_1^1 = \frac{2(x_0' - x_f')\tau_f + (x_0'' + x_f'')\tau_f^2}{4\pi}, \ b_2^1 = \frac{12(x_0 - x_f) + 6(x_0' + x_f')\tau_f + (x_0'' - x_f'')\tau_f^2}{24\pi} \tag{5.2.24}$$

for the x-coordinate case and similar for the y-coordinate case. The task is now to find the optimal solution among all the trajectories described by the equations, by varying the parameter τ_f.

The trajectory is described by a set of N evenly spaced points on a virtual arc $[0;\tau_f]$ with the interval

$$\Delta\tau = \tau_f (N-1)^{-1} \tag{5.2.25}$$

so that

$$\tau_j = \tau_{j-1} + \Delta\tau \ , \quad j = 2, ..., N \ , \ (\tau_1 = 0) \tag{5.2.26}$$

which is substituted into Equations (5.2.20) to (5.2.22) to define the x coordinates and their derivatives (and similar for y) for each point on the trajectory.

For each node, $j = 2, ..., N$, the following is computed:

$$\Delta t_{j-1} = \sqrt{\frac{(x_j - x_{j-1})^2 + (y_j - y_{j-1})^2}{V_h^2 + W^2 - 2V_h W \cos\psi_{j-1}}} \ , \ (\psi_1 \equiv \psi_0) \ , \tag{5.2.27}$$

$$\lambda_j = \Delta\tau \Delta t_{j-1}^{-1} \ , \tag{5.2.28}$$

$$\psi_j = \tan^{-1} \frac{\lambda_j y_j'}{\lambda_j x_j' + W} \ , \tag{5.2.29}$$

$$\dot{\psi}_j = (\psi_j - \psi_{j-1}) \Delta t_{j-1}^{-1} \ . \tag{5.2.30}$$

Finally, the trajectory performance index is calculated as

$$J = \left(\sum_{j=1}^{N-1} \Delta t_j - T_{turn} \right)^2 + k^{\dot{\psi}} \Delta \tag{5.2.31}$$

where

$$\Delta = \max_j(0; |\dot{\psi}_j| - \dot{\psi}_{jmax})^2 \tag{5.2.32}$$

and $k^{\dot{\psi}}$ is a weighting coefficient [8].

An optimisation function is now employed to find the value of τ_f for which the cost function J is minimised. After finding the optimal τ_f, the optimal TG trajectory is described by N points, each with a position (x,y) and derivatives, heading ψ and yaw rate $\dot{\psi}$.

The cost function in Equation (5.2.31) penalises trajectories for which the time duration is not equal to the desired turn time, T_{turn}, and which require yaw rate commands that exceed the maximum allowed yaw rate command.

Any optimisation algorithm can be used to find the optimal τ_f value. A simple linear search over an appropriate range of values is deemed adequate. N, the number of path nodes, is chosen arbitrarily to provide high enough resolution for the path, while being low enough to maintain a short execution time. The value of 20 gives adequate results.

5.2.5 Trajectory Analysis

The optimal TG algorithm can generate trajectories for a range of initial conditions. When the radius for the turn becomes too small, the generated optimal trajectory requires yaw rate commands that exceed the desired maximum of 20°/s. The minimum radius is determined as 66 m, but leaves no room for correction for initial position errors. A more suitable minimum radius of 80 m is used. Figures 5.9 to 5.11 show generated trajectories for various initial conditions. As seen in Figure 5.9, feasible trajectories that fall within the allowable yaw rate range can be planned for an altitude error of up to 7 m below the ideal altitude. Figure 5.10 shows trajectories with the same starting altitude, but with different horizontal offsets from the ideal initial position. The shown results are the maximum offsets in their respective directions, while still adhering to the specified yaw rate limit and completing the turn in exactly the available time. Similar, Figure 5.11 shows trajectories for the maximum heading errors at the ideal initial position.

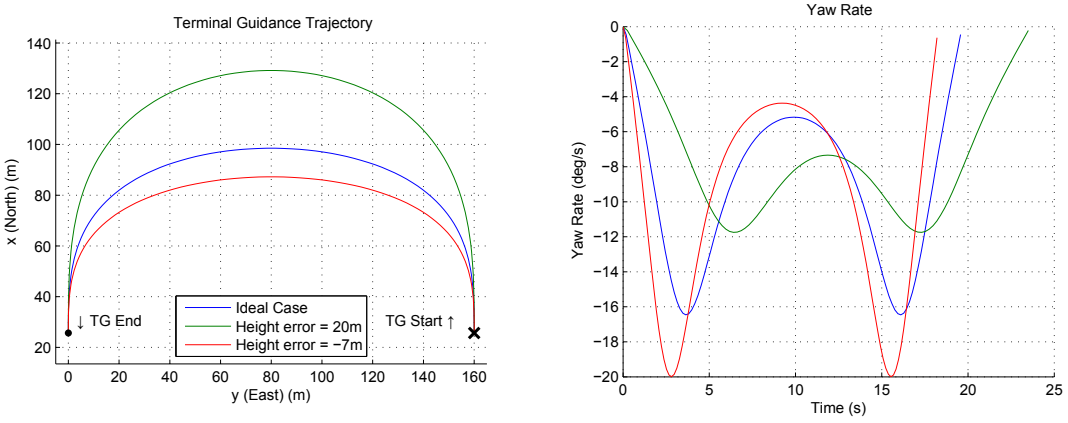


Figure 5.9 – Optimal terminal guidance results for different initial positions varying in altitude, $R = 80$ m.

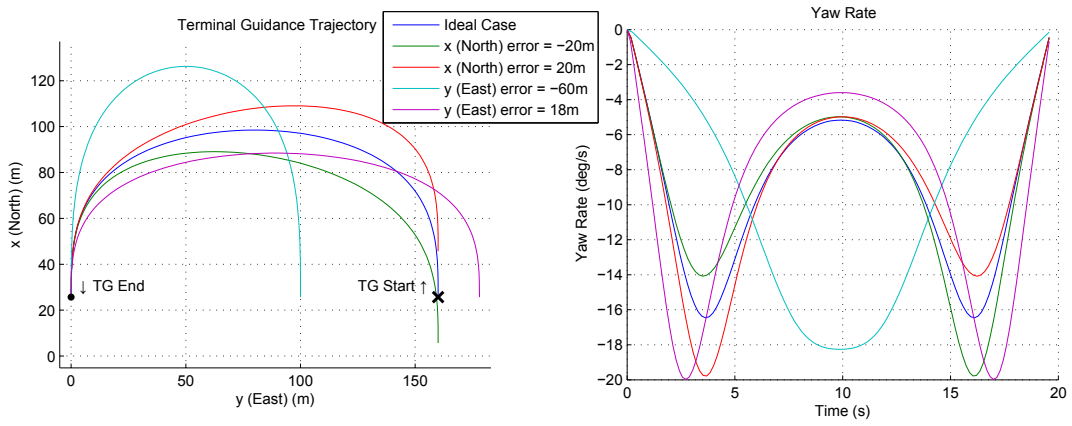


Figure 5.10 – Optimal terminal guidance results for different initial positions varying in horizontal offset, $R = 80$ m.

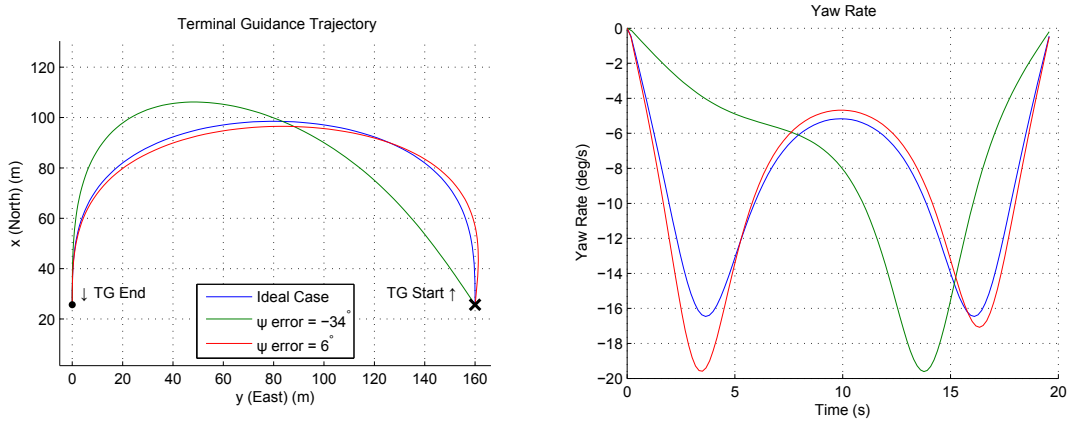


Figure 5.11 – Optimal terminal guidance results for different initial conditions varying in heading angle, $R = 80$ m.

5.2.6 Trajectory Following

The generated optimal TG trajectory can be followed with a few methods:

- The **yaw rate controller** of Section 4.3 can be used to track the calculated yaw rate command. Yakimenko and Slegers [8] use a model predictive yaw rate controller to track the yaw rate command with simple additional tweaks to enhance the tracking performance. The algorithm is replanned during the phase to compensate for tracking errors. However, it is found that pure yaw rate tracking with the yaw rate controller developed for this project is inadequate. Firstly, the actual yaw rate will lag any yaw rate command due to the response of the yaw rate controller and plant. This can be remedied by commanding the controller a few seconds earlier. In this way, the required yaw rate response can be obtained. However, the simple 2D kinematic model used by the optimal TG algorithm differs from the parafoil model, causing the actual flown path to deviate from the planned path, even if the yaw rate response is the same. Further, sharp yaw rate commands produced by the algorithm are hard to follow and small deviations from this cause a large deviation in path. For this reason,

even replanning the trajectory en route is not adequate.

- The reference trajectory can be followed by the **cross-track path follower** of Section 4.5.1. The algorithm's yaw rate command is provided as the yaw rate controller feed forward while the calculated path is used for the cross-track error. It is easy to understand that this method is an enhancement over pure yaw rate tracking.
- Lastly, the **non-linear guidance method**, introduced in Section 4.5.2, can be used to follow the reference trajectory. This guidance method naturally takes the path curvature into account and with a sufficiently small reference point distance, L_1, the path is followed more accurately than the previous two methods and without the need for additional tweaks.

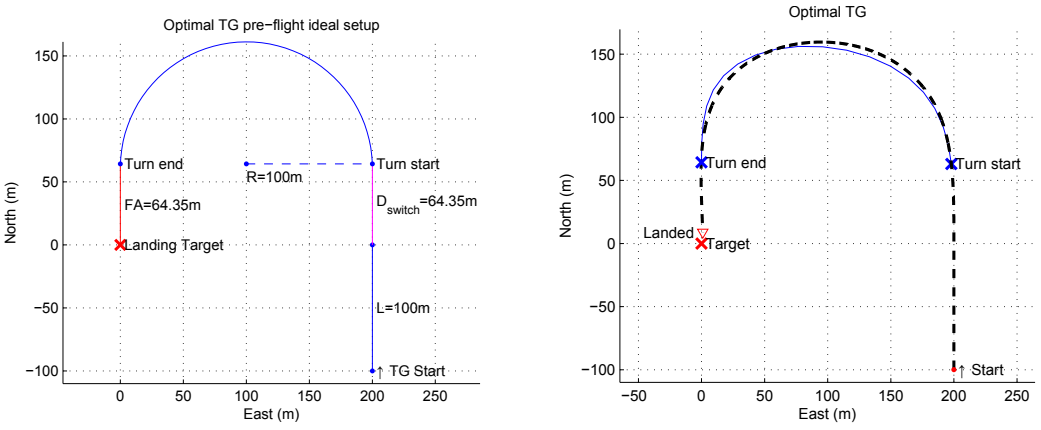
The non-linear guidance method is chosen as the tracking method for the optimal guidance trajectory since it is the most straight forward to apply to the problem and produces best tracking performance for the optimal TG trajectories.

The non-linear guidance method depends on a reference point on the desired path, some distance ahead of the vehicle. Since the optimal TG trajectory is only generated when the vehicle reaches the appropriate location, the guidance method will be set to follow the constant turn circle connecting the ideal TG boundary conditions until the optimal trajectory is generated.

Arriving at the TG turn start with excess height is advantageous since the resulting optimal trajectory will have a more gradual curvature change which can be tracked more closely. Using a smaller value for L_1 also leads to more exact path following in the TG turn.

5.2.7 Simulation

The optimal TG phase is simulated with the non-linear parafoil model. The ideal boundary conditions are set up as shown in Figure 5.12(a) with a turn radius $R = 100$ m, $L = 100$ m and FA time of 5 s. The simulation results with the non-linear guidance method are shown in Figure 5.12(b). The parafoil lands 9 m before the desired landing target.



(a) Preflight ideal TG setup.

(b) TG simulation, $L_1 = 100$ m

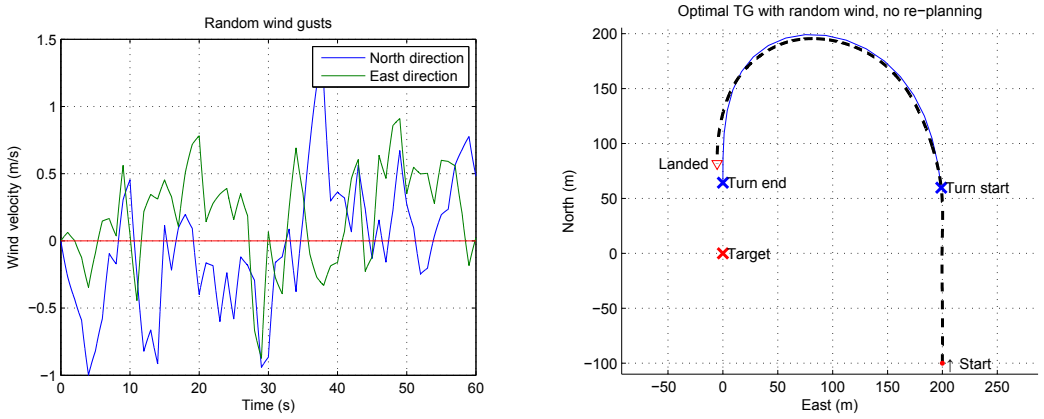
Figure 5.12 – Optimal TG ideal setup and simulation.

Due to differences in the actual velocities from the values used for generating the trajectory, as well as the fact that the simple kinematic model used differs from the actual vehicle model,

the vehicle might not reach the landing target exactly, even if the trajectory following is accurate. This can be accounted for by replanning the trajectory en route. Even replanning only once, midway through the turn, increases the final position accuracy. In addition, replanning will also compensate for external errors such as unknown wind.

To illustrate the effect of replanning under windy conditions, the simulation is run with randomly generated wind gusts, shown in Figure 5.13(a). For the first case, with no replanning, the parafoil follows the optimal trajectory that was generated at the turn start, shown in Figure 5.13(b). The flown trajectory of the parafoil is represented by the black dashed line. The non-linear guidance method is robust against the unknown wind and maintains adequate path tracking. However, due to the wind, the parafoil reaches the ground 82 m before the desired landing target.

Setting the system to replan the optimal TG trajectory en route every 2 s produces better results, enabling the parafoil to land 5 m from the target location under the same wind conditions, as shown in Figure 5.14. The positions of the vehicle at the replan instances are denoted with X's.



(a) Random generated wind gusts.

(b) Optimal TG simulation without replanning.

Figure 5.13 – Optimal terminal guidance simulations with random wind gusts.

5.3 Homing

The trajectory for the homing phase consists of Dubins paths from the launch point, intersecting optional intermediate waypoints (if any) and ending at the start point of the EM phase, as illustrated in Figure 5.15. After the launch, the parafoil is allowed to settle by flying a straight leg in the launch direction, which ends at the first waypoint. If the following waypoint is an intermediate waypoint (i.e. not the EM start point), only an initial turn and a tangent connecting the turn and the next waypoint is generated, since it is not required to arrive at an intermediate waypoint with a specific heading. This is repeated until the following waypoint is the EM start point. For the last section, both an initial and final turn are constructed and joined with a tangent line in order to respect the required heading of the EM start point.

Figure 5.16 graphically depicts algorithm output between two nodes for each case. Figure 5.16(a) illustrates the case when the following waypoint is an intermediate waypoint. In this case, only the first waypoint possesses a heading (at which the vehicle arrived at the waypoint), while the second waypoint does not have a predetermined heading and can thus

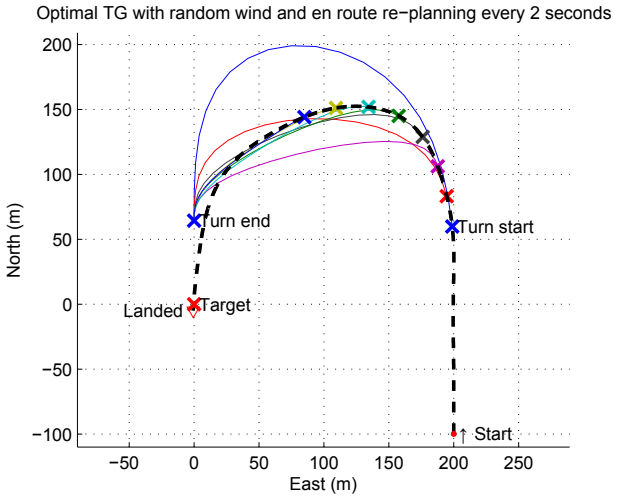


Figure 5.14 – Optimal terminal guidance simulation with random wind gusts, with en route replanning every 2 s.

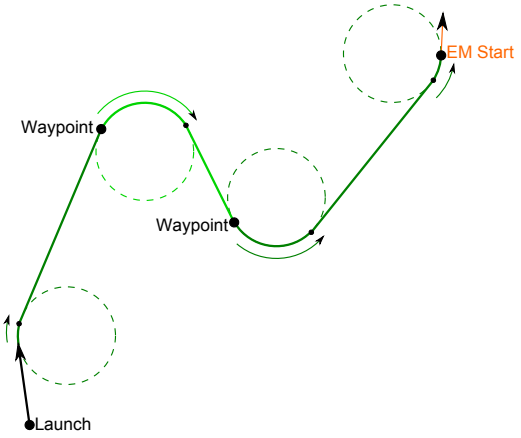
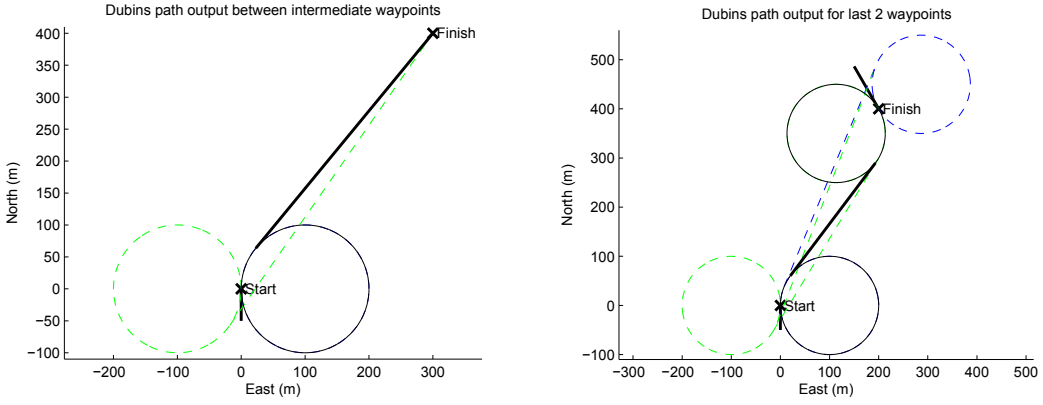


Figure 5.15 – Trajectory of the homing phase.



(a) Path to an intermediate node.

(b) Path to the last node (EM start).

Figure 5.16 – Dubins path algorithm output.

be approached from any direction. Two turning circles can be created, on the left and right-hand side of the first waypoint, tangential to the heading direction and intersecting the waypoint. From this, two valid paths are possible, connecting each turning circle to the next waypoint. The path with the shortest turn is chosen.

Figure 5.16(b) illustrates the last case where the current waypoint is to be connected to the EM start point. Both waypoints have a desired heading and left and right turning circles are constructed for both waypoints. In this case, 4 valid paths are available, as shown in the figure. Again, the shortest path is chosen.

Although the trajectory follower is able to transition between two straight legs that intersect at an angle, without the need for a predetermined turning circle, the use of explicit turning circles allow the length of the path, and thus the altitude required, to be determined exactly.

The altitude required for the path is calculated by

$$z_{homing} = \frac{D}{V_h} V_v \tag{5.3.1}$$

where D is the total distance covered by the path, calculated with basic geometry, and constant values are assumed for the velocities V_h and V_v.

To ensure that a valid path can always be generated, and for the sake of simplicity, contiguous waypoints are required to be a distance of more than 2 turning circle diameters apart, or $4R$ where R is the turning circle radius.

In the event that the parafoil is released directly above the landing target, the task falls upon the user to command at least one waypoint sufficiently far enough away from the start and landing point to enable the path planning algorithm to construct a valid path.

5.4 Energy Management

The number of EM circles required are calculated based on the remaining flight altitude and the altitude loss expected from a single circle, assuming that V_v and V_h remain constant. For a radius R_{em}, which can be any chosen value larger than the minimum turn radius of the parafoil, the altitude loss for a single circle is given by

$$z_{circ} = \frac{2\pi R_{em}}{V_h} V_v \tag{5.4.1}$$

$$= kR_{em}, \quad k = \frac{2\pi V_v}{V_h} \ . \tag{5.4.2}$$

The number of circles required for the EM phase is

$$c = \frac{(h_{ems} - h_{tgs})}{z_{circ}} \tag{5.4.3}$$

where h_{ems} and h_{tgs} are the heights at the start of the EM and TG phases, respectively. It is required that the number of circles be an integer. If c is not an integer, the turn radius is enlarged so that an integer number of turns will result in the required amount of altitude loss. The integer number of turns is

$$n = \text{floor}(c) \ . \tag{5.4.4}$$

Thus, if only n turns are allowed, the new height required for a circle is (assuming $n > 0$)

$$z_{circ2} = \frac{(h_{ems} - h_{tgs})}{n} = kR_{new} \tag{5.4.5}$$

where R_{new} is the new required radius, which is finally calculated as

$$ckR_{em} = (h_{ems} - h_{tgs}) \tag{5.4.6}$$
$$= nkR_{new} \tag{5.4.7}$$
$$\Rightarrow R_{new} = R_{em}\frac{c}{n} \tag{5.4.8}$$

During flight, before entering each circle, the required radius is calculated again using the current height and velocity measurements, thereby compensating for disturbances and inaccuracies due to assumptions.

It is advantageous to choose R_{em} larger than the minimum turn radius of the parafoil to allow the path follower to correct for disturbances without saturating the yaw rate command.

5.4.1 Using EM for Wind Estimation

An added advantage of the EM phase consisting of circles is that it is ideally suited for a method of wind estimation described by [12, p. 20]. By flying a constant turn and measuring the displacement after a full turn, an estimate of the average wind is calculated. This knowledge can then be used to improve the terminal guidance phase.

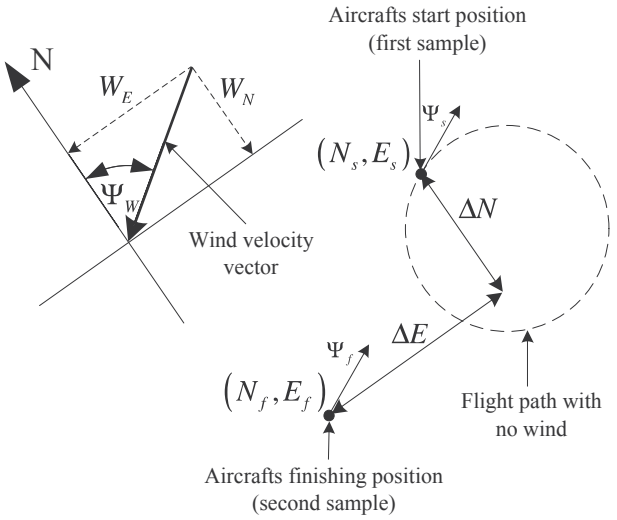


Figure 5.17 – Wind estimation procedure (from [12, p. 21])

The procedure is illustrated graphically in Figure 5.17. An open loop constant turn is initiated and maintained. Initial position and heading measurements are taken when the vehicle has settled into the constant turn. The initial heading is reached again when a full turn is achieved and the final measurements are taken. The displacement due to the wind is given by

$$\Delta N = N_f - N_s \tag{5.4.9}$$
$$\Delta E = E_f - E_s \tag{5.4.10}$$

allowing the north and east wind velocity components to be calculated as

$$W_N = \frac{\Delta N}{t}\,, \qquad W_E = \frac{\Delta E}{t} \tag{5.4.11}$$

where t is the time duration of the turn. The wind heading is calculated as

$$\psi_w = \tan^{-1}(\frac{\Delta E}{\Delta N}) \ . \tag{5.4.12}$$

Since the procedure will result in a displacement from the initial start point, there are some practical considerations involved. It is advantageous to choose a turn radius as small as possible so that the resulting drift is minimised. The displacement due to a 2 m/s wind is approximately equal to the circle radius (for the velocities of the current parafoil model). In the worst case, the wind would be in the final landing direction (the negative x-direction in the optimal TG axes). This would cause the vehicle to be offset the distance of the circle radius further away from the TG phase start. The parafoil now has to travel the extra distance in addition to moving against the wind. This scenario is illustrated in Figure 5.18. The EM wind estimation turn is depicted by the black dashed line. For a circle radius of 100 m, the wind of 2 m/s causes a displacement of 100 m, thus leading to a lengthened L. Using the optimal TG formulae to generate the appropriate trajectory, this amounts to an additional altitude of 85 m required in order for the vehicle to complete the TG phase and reach the landing target.

Thus, in order to use this wind estimation technique, the extra height for the worst case wind has to be taken into account when planning the trajectory.

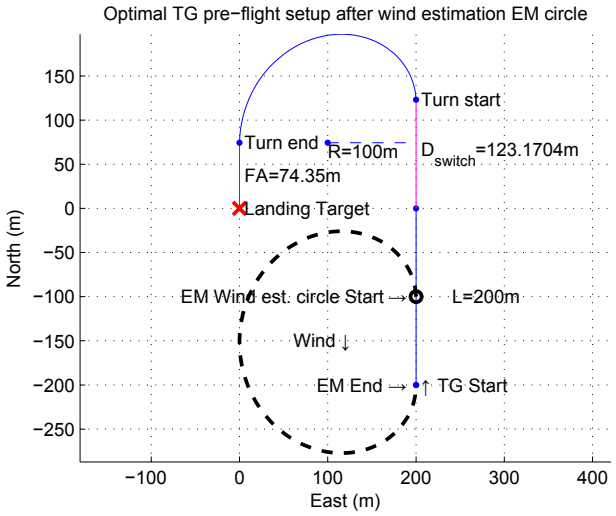


Figure 5.18 – Optimal TG setup after EM wind estimation routine for 2 m/s wind in southern direction.

5.5 Mission Control Summary

This chapter outlined the path planning for various flight phases.

The optimal TG algorithm by [8] is chosen as the preferred TG method as it has several advantages over the modified Dubins method, including:

- Initial heading and turn rate errors are accounted for, in addition to position errors.
- The generated trajectory has a smooth yaw rate and heading transition.
- A constant, known wind can be taken into account.

- The trajectory can easily be replanned from any point, accounting for unknown wind, measurement errors and unmodeled dynamics.

The optimal TG algorithm makes provision for a constant known wind in the landing direction. Section 5.4.1 discussed a simple method of obtaining a wind estimate from an open loop EM turn. However, this method is regarded as too expensive in terms of the extra altitude required. Wind estimation techniques are left as a recommendation for further research and the wind is regarded as unknown disturbances which is compensated for by the rapid replanning of the optimal TG algorithm en route.

The following parameters need to be chosen for the path planning algorithms:

- Distance after take-off, before the start of the homing phase. This may be chosen arbitrarily to allow the parafoil to settle into straight flight.
- The turning radius for homing, EM and TG turns. The values do not have to be the same for the different phases, and have to be large enough as not to exceed the maximum allowed yaw rate command. A value of 100 m for all phases is deemed adequate.
- The FA time or distance should be chosen to allow for enough time for the parafoil to settle into straight flight before the flare manoeuvre is performed.

At the end of the FA, prior to landing, the flare manoeuvre takes place. As shown in Section 2.4.2, the manoeuvre reduces the forward and downward velocities leading to a softer touchdown. The timing of the manoeuvre is very precise and requires careful consideration. Designing an accurate flare manoeuvre is outside the scope of this project and is recommended as future research, as well as the addition of an accurate relative ground height sensor.

Logic is implemented to ensure that a valid path is planned preflight, given the start and landing points, the available altitude and the expected velocities. If, during flight, the situation arises that insufficient altitude is available to complete the desired path, different strategies can be implemented to react to the situation. Firstly, if the current phase is TG, the TG algorithm will automatically attempt to take care of the problem. If the current phase is not TG, but the vehicle is in an appropriate position to initiate a TG phase, the TG algorithm is used to plan a trajectory to the landing point. Finally, if the current phase is the homing or EM phase, the vehicle is not in an appropriate position for a TG phase and reducing the excess EM circles is not sufficient, two choices are considered:

- The parafoil can turn and head directly to the landing target in an attempt to land as close as possible.
- The parafoil can simply continue the homing phase and land en route when the ground is reached.

The latter option is chosen since it will ensure that the parafoil lands somewhere along the preplanned route. If waypoints were selected by the user with intention to avoid dangerous areas, the former option may endanger the vehicle as it could land far off route in an unwanted area. Additionally, landing somewhere en route could allow the vehicle to be found easier if communication was lost and the landing location is unknown.

Chapter 6

Avionics Hardware and Software

For this project, an avionics system which has been developed at the ESL for a previous project [27] is used. The avionics system has been used extensively for past UAV projects at the ESL, making it a logical choice of avionics to use, eliminating the need to design or purchase a new system. In addition, the avionics system had already been adopted by the UCT team who had purchased the system for the aim of logging flight test data. Subsequently, the work of this project was integrated into the existing system. The avionics system is housed in the payload box along with the UCT steering unit which includes the brake line winches, servo controller and batteries. Figure 6.1 contains an annotated photo of the open payload unit.

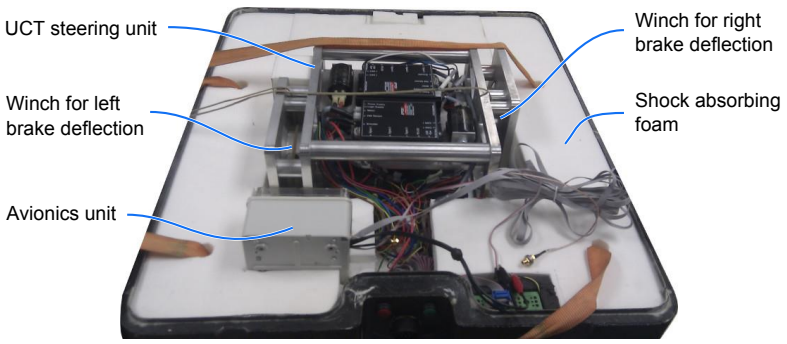


Figure 6.1 – Open payload box displaying the avionics and steering unit.

The complete autonomous parafoil system is the result of an integration of components from the current project and the UCT team. For clarity, the components which are designed by the respective teams are distinguished by referring to an *ESL* component for a component designed by the ESL or the current project and a *UCT* component as a component designed by the UCT team, with which the current project interfaces.

This chapter provides an overview of the avionics system and discusses the project specific changes made as well as the integration with the UCT system.

6.1 Avionics Overview

The standard ESL avionics system consists of various nodes connected on a CAN (Controller Area network) bus, including:

- On-board computer (OBC)
- IMU (Inertial Measurement Unit) node
- Servo controller node

The central component of the avionics system is the OBC which contains:

- Two 16 bit dsPIC30F6014 microcontrollers (denoted A and B)
- u-Blox RCB-4H GPS receiver
- MaxStream X24-019NM OEM RF module
- SD card slot with data logging functionality

Microcontroller A serves as the main node on the CAN bus. A servo controller board, which is also developed at the ESL, is usually also connected as a node on the CAN bus as listed above. However, for this project, the OBC is required to communicate with a servo controller board developed by the UCT team which does not provide functionality for connecting to a CAN bus. The only free unused communication method was determined to be a RS232 serial port on microcontroller B and the UCT servo controller board. Microcontroller B is thus employed to function as the traditional servo controller CAN node, handling communication to and from the UCT servo controller.

The IMU node contains the sensors used which include:

- 3× Analog Devices ADXRS613 ±150°/s Yaw Rate Gyroscope.
- 2× Analog Devices ADXL210E Low-Cost ±10g Dual-Axis Accelerometer
- 1× Honeywell HMC2003 Three-axis magnetometer

The IMU node is discussed in detail by [28]. An additional IMU node, containing gyroscopes and accelerometers, is fixed to the parafoil canopy, by which the canopy movement relative to the payload can be determined.

A functional overview of the OBC components unique to this project are given in the following subsections. A complete, in-depth description of the OBC and its components can be found in [27].

6.2 Microcontroller A

Microcontroller A performs the following functions:

- Controls the GPS module
- Receives sensor and GPS data and updates a kinematic state estimator
- Handles communication with the ground station
- Maintains the SD card file system and performs data logging
- Runs the flight control system (FCS) and outputs actuator commands on the CAN bus

The kinematic state estimator was originally designed in the ESL by [29] and implemented and used successfully in a number of ESL projects. Additional changes made over time are documented by [30].

Communication with the ground station is done using the protocol defined by [18, p. 159]. Periodic telemetry packets are sent by the OBC to the ground station while commands are received from the ground station. The particular frequencies of the various telemetry packets are not of critical importance and can be changed to the user's preference in the source code as long as the bandwidth and maximum data rate of the RF connection is kept in mind. To save bandwidth, only a selection of the control system related telemetry is sent per packet depending on the current controller state, which is either *setup*, *flight planning* or *flight*, corresponding to the flight setup process described in Section 6.4.

The data is logged to the SD card at a rate of 25 Hz using a FAT32 file system. The data file allocation is implemented such that a sudden loss of power will not corrupt the file system or logged data. The data is extracted post flight with the aid of MATLAB scripts.

The FCS is implemented on microcontroller A and runs at a rate of 50 Hz. The implementation is discussed further in Chapter 7.

6.3 Microcontroller B and Servo Communication

Brake deflection is performed by the servos and commanding the servos is facilitated by the UCT microcontroller. The actuators can either be controlled by the safety pilot (SP) or by the autopilot (AP), determined by two AP switches. The block diagram in Figure 6.2 provides an overview of the SP and AP structure and depicts the role of microcontroller B.

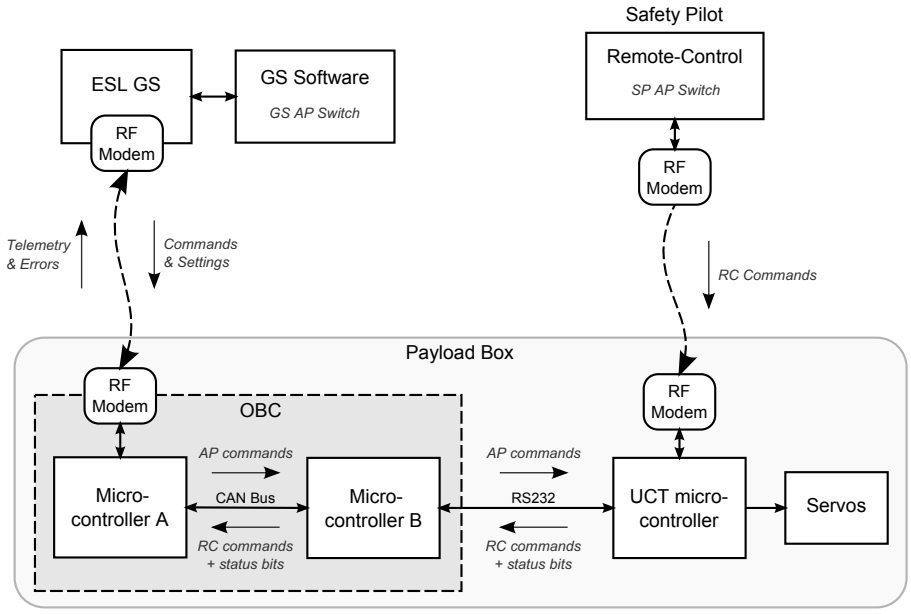


Figure 6.2 – Block diagram of servo communications.

Microcontroller B acts as the servo controller CAN node, replacing the conventional ESL servo controller board. From the ESL side, microcontroller B receives actuator commands from microcontroller A at a rate of 50 Hz via the CAN bus and passes it to the UCT

microcontroller over a RS232 connection. These commands are the FCS output values if the AP is armed, or zero otherwise. From the UCT side, microcontroller B receives the remote control (RC) commands from the SP and passes it to microcontroller A via the CAN bus to be logged to the SD card.

As seen in Figure 6.2, two AP switches are present – one at the SP and one at the ESL ground station. The parafoil is under the manual control of the SP when the AP switch on the SP side is unarmed. In this case, the UCT microcontroller ignores any commands received from microcontroller B and only listens to the RC commands. When the SP arms the AP from his side, the UCT microcontroller starts listening to the commands from microcontroller B, ignoring the RC commands. At this stage, if the AP switch at the ESL ground station is unarmed, the UCT microcontroller will be receiving zero commands. Only once the AP at the ESL ground station has also been armed, microcontroller A will transmit the AP commands.

The communication protocol used by the RC RF link on the UCT side was defined by A. Grunwald on the UCT team [17] and had to be implemented on microcontroller B in order for the two systems to be compatible. The protocol defines unique start-of-packet and end-of-packet characters with data containing bytes and one checksum byte in between. In order to prevent one of the data bytes coincidentally matching the value of the start or end character, the data and checksum bytes are broken up into 6-bit words and restructured. This means that the restructured bytes are each restricted to a maximum value allowed by 6 bits, while the start and stop characters may use 8 bits. Choosing the start and stop characters as values exceeding the 6-bit maximum ensures that the data and checksum containing bytes will never equal the start or stop characters. This method is referred to as the base-64 method.

The data contained in each packet is given in Table 6.1. Note that this refers to the sequence of data bytes within the servo command package before it is processed and converted to the protocol-specific package. The first 4 bytes contain the left and right brake deflection commands. Bytes 4 to 7 represent trim deflection settings to which the brakes can be set independent from the deflection commands. Each of the first four table entries are represented by a signed 16-bit integer, requiring 2 bytes of storage each. Byte 8 contains status bits related to the servo controller and is described in Table 6.2. For this project, only status bit 5 is of interest, which indicates the state of the AP switch at the SP side.

Byte	Description	Range
0-1	Left servo command	-2546 to 2546
2-3	Right servo command	
4-5	Left trim	
6-7	Right trim	
8	Status bits	N/A

Table 6.1 – Data format of servo command package.

Bit	Function	Description
(LSB) 0-4	Reserved	
5	AP switch	1=RC (AP unarmed, manual SP control) 0=AP (AP armed from SP side)
6-7 (MSB)	Reserved	

Table 6.2 – Servo command status byte package.

As seen in Table 6.1, the servo commands and trim values are required to be within the specified range, where a value of 0 corresponds to zero brake deflection and a value of ±2546 corresponds to ±100% brake deflection, respectively. This means that brake deflection commands can be negative which can be used for a case where the trim requires a negative offset. The maximum value is a result of the steering line length, winch diameter, motor gearing ratio and shaft encoder [17]. Using the maximum value of 2546 means that the minimum achievable brake deflection value is

$$\delta_{min} = \frac{100\%}{2546} = 0.039\% \ . \tag{6.3.1}$$

If a yaw rate to brake deflection gain of roughly 2 is assumed, the minimum achievable yaw rate is approximately 0.02°/s which is below the expected gyroscope noise level and deemed acceptable.

Infrequent access was available to the UCT system. In order to test the servo communications implementation, a computer application was created which could communicate with microcontroller B via a serial port. Using the same communications protocol, the application can act as the UCT microcontroller, modify the RC commands and status bits and even pass the received servo commands on to a Simulink simulation using a TCP connection. This allowed thorough testing of the implementation resulting in a seamless integration with the UCT system.

6.4 Ground Station Software

The ESL ground station (GS), depicted in Figure 6.2, consists of a GS hardware unit and the GS software running on a PC. The GS hardware unit connects to the PC running the GS software via a serial port and is used to send and receive commands and telemetry data between the GS software and the OBC. The GS software enables the user to control certain behaviour of the OBC and provides the following functionality:

- Reset and initialise the OBC
- Format the SD card and start or stop on-board logging
- Initialise the on-board state estimator
- Receive OBC status information and error messages
- View IMU, GPS and estimator telemetry data received from the OBC
- Arm and disarm the autopilot
- Logs all received and sent data to files on the PC
- View the status bits and commands received from and sent to the servo controller
- View and set up controller parameters and flight plan
- Stream telemetry data to another application via a TCP/IP connection

The GS software is developed in the ESL and the basic framework with functionality of the first seven points listed above was available to start with. This greatly reduced development time since only the project specific functionality needed to be added to the existing basic GS framework.

The GS software is logically structured into multiple tabs which are discussed in the following sections.

6.4.1 Main Window

The default main window of the GS software at startup is shown in Figure 6.3. The default window contains the *Main* sidebar on the left, a centre area in which the various tabs are located, a right sidebar and a status bar at the bottom. The *Main* sidebar is used to control the basic behaviour of the OBC, including reset and initialisation, SD card logging control, estimator and autopilot control, selection of the serial port to which the GS hardware unit is connected and a *Command History* text box.

In the centre area of the window, the *Console* tab is shown as the current active tab, containing a large text area which is a mirror of the *Command History* in the *Main* sidebar. The *Command History* area displays a history of all the commands performed and any error or warning messages or confirmation messages from the OBC. Additionally, functionality to log a custom message is included, enabling the user to add custom notes for later reference. The entire contents of the *Command History* is logged to a file, providing a convenient record of the events for future reference.

The status bar along the bottom of the window provides fast, convenient access to the current status of various OBC components.

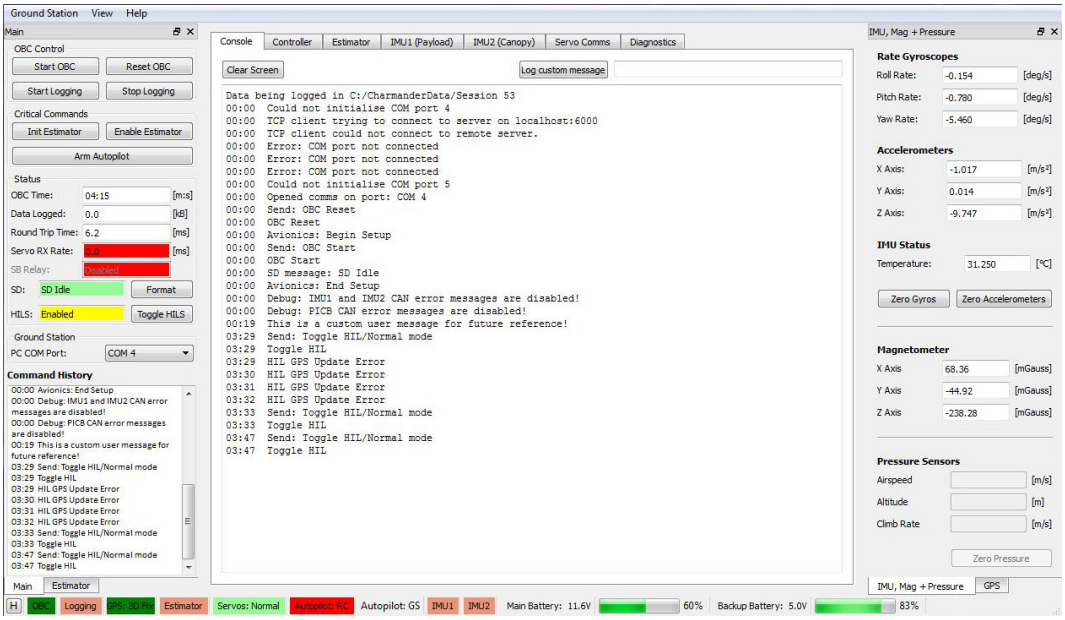


Figure 6.3 – GS main window.

6.4.2 Controller Tab

The *Controller* tab groups all of the FCS related functionality and is divided into *Setup*, *Flight Planning* and *Flight* tabs which represent the various flight test stages and aims to guide the user in a logical manner when preparing the OBC for a flight test.

Figure 6.4 shows the *Setup* tab. Parameters of the various FCS components are displayed and can be modified by the user. The different controllers can be enabled or disabled, and the user can select a flight type from a dropdown box. The flight type determines the in-flight behaviour when the AP will be armed and is described in Table 6.3. Initially, the GS is in *Setup* mode and the *Flight Planning* and *Flight* tabs contents are disabled. Only once

Flight type	Description
Step command	Use user-defined command as step input to controller
Optimal TG only	Only perform optimal TG manoeuvre
Full flight	Perform full flight with homing, EM and TG phases

Table 6.3 – GS selectable flight types.

the user has clicked the *Activate* button on the *Setup* tab will the next tab be activated. If the flight type is either of the latter two listed in Table 6.3, clicking the *Activate* button will enable the *Flight Planning* tab. If the *Step command* flight type is selected, the GS skips to the *Flight* tab.

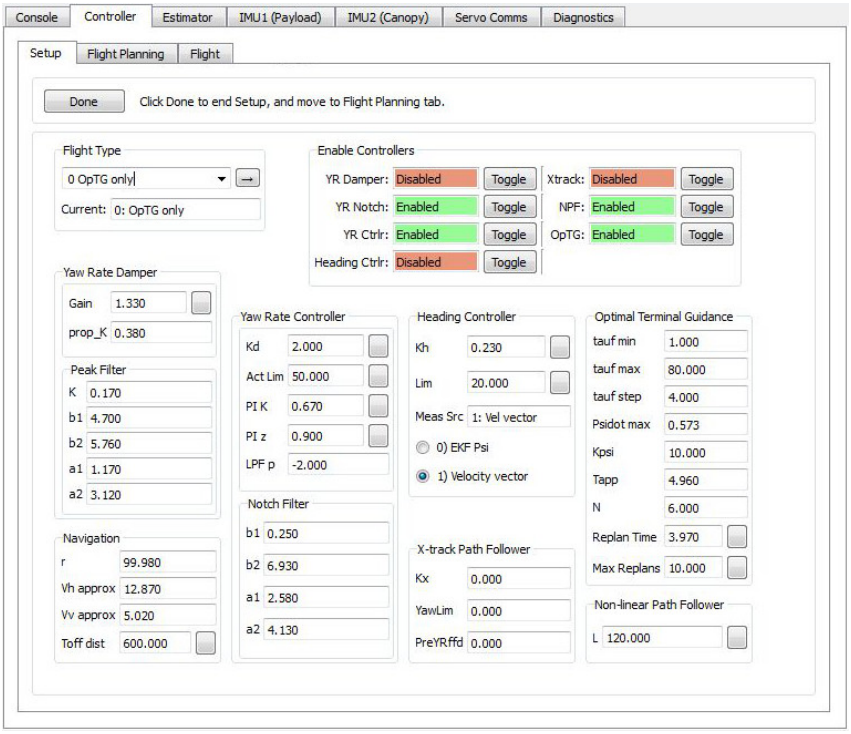


Figure 6.4 – GS controller setup tab.

Figure 6.5 shows the *Flight Planning* tab which allows the user to set the landing target, add waypoints and review the planned flight. A summary is displayed, providing information on the number and size of EM circles required or which warns the user if insufficient height is available to reach the landing target. Once a valid path has been planned, the user can click the *Done* button to progress to the *Flight* tab.

The *Flight* tab is shown in Figure 6.6. On the left-hand side, a *Status* text box lists current status information on the FCS. Initially, a summary of the planned flight is displayed. As the flight progresses, the current FCS state is displayed. Information is appended to the text box throughout the flight, enabling the user to scroll back and review the FCS history. On the right-hand side, information relevant to the current flight type is displayed graphically. For *Full flight* and *Optimal TG only* flight types, the planned and flown trajectory is displayed, as well as the vehicles current position.

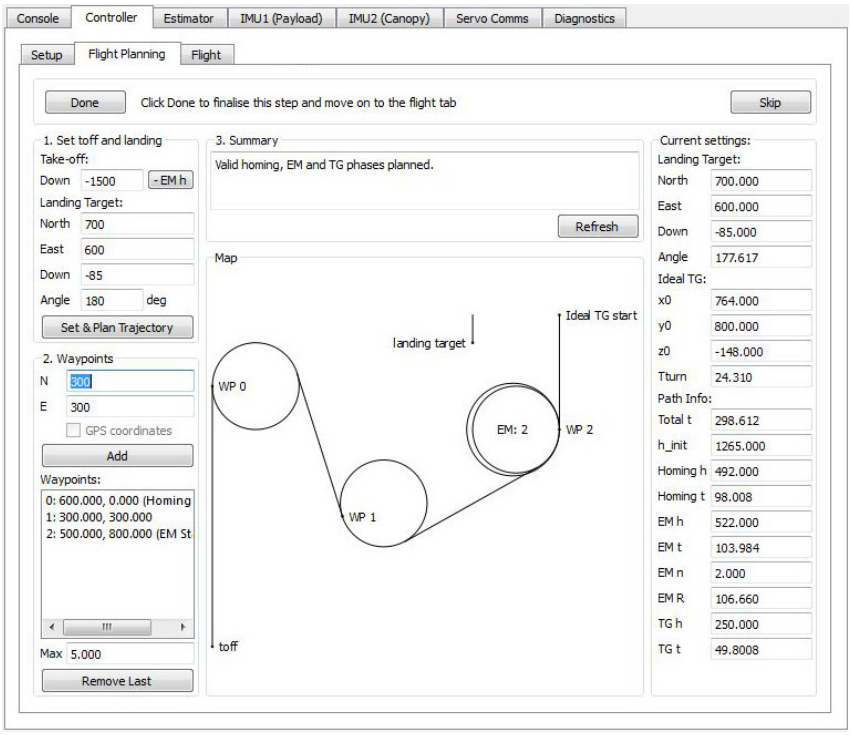


Figure 6.5 – GS controller flight planning tab.

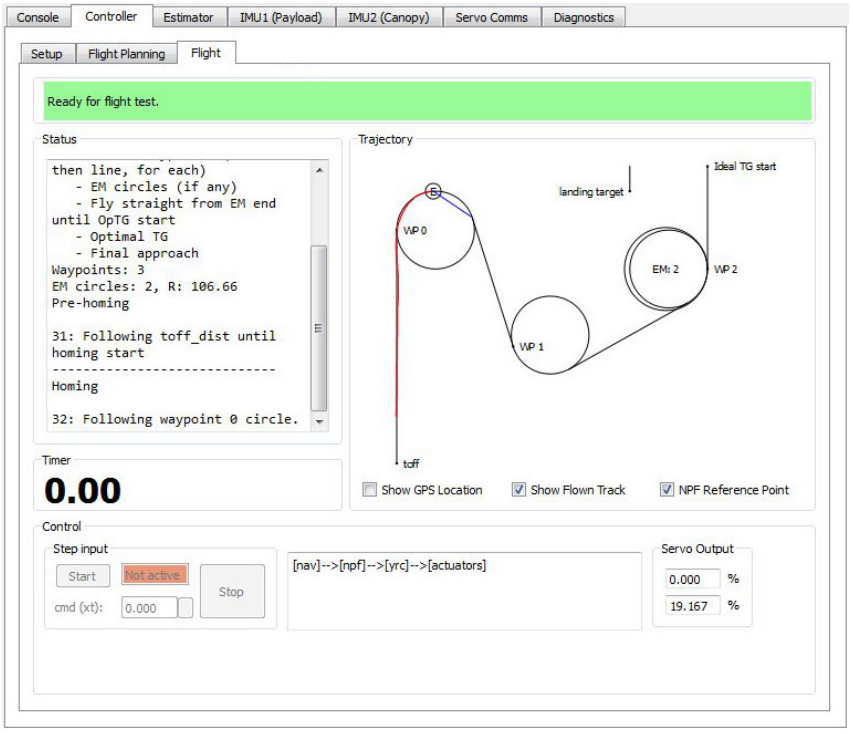


Figure 6.6 – GS controller flight tab during a full mission.

The *Step command* flight type allows the user to activate a step input to the enabled controllers. When the AP is armed, clicking the *Start* button in the *Step input* group box initiates a step command to the outer-most enabled controller. If a path follower controller is enabled, the step command is interpreted by commanding the path follower to track a straight line in the direction of the current heading which is offset laterally at a distance equal to the magnitude of the step command. As the step command is initiated, a timer is started, displaying the duration of the step to the user. The step response is also plotted in the graphical view on the right-hand side of the tab as seen in Figure 6.7 which shows the *Flight* tab during a step input command.

Figure 6.7 – GS controller flight tab during a step input command.

6.4.3 Estimator and IMU Tabs

The *Estimator* and *IMU1 (Payload)* tabs were inherited from the original GS framework and *IMU2 (Canopy)* was added for this project. The tabs display telemetry information related to the respective components. On the *Estimator* tab, the state estimator can be initialised and enabled.

6.4.4 Servo Comms Tab

The *Servo Comms* tab is shown in Figure 6.8. On the right-hand side of the tab, the servo commands being received from and sent to the UCT microcontroller are displayed in the top and bottom group box, respectively. On the left-hand side, functionality is provided to control the servos manually. This is useful for testing the servo communications before a flight. The servos can only be controlled manually if the AP is armed from the SP side

and manual servo control is enabled in the GS by clicking the *Enable Manual Servo Control* button. For test cases when the RC is not available, functionality is provided to manually force the SP AP flag on the OBC in order to test servo communication.

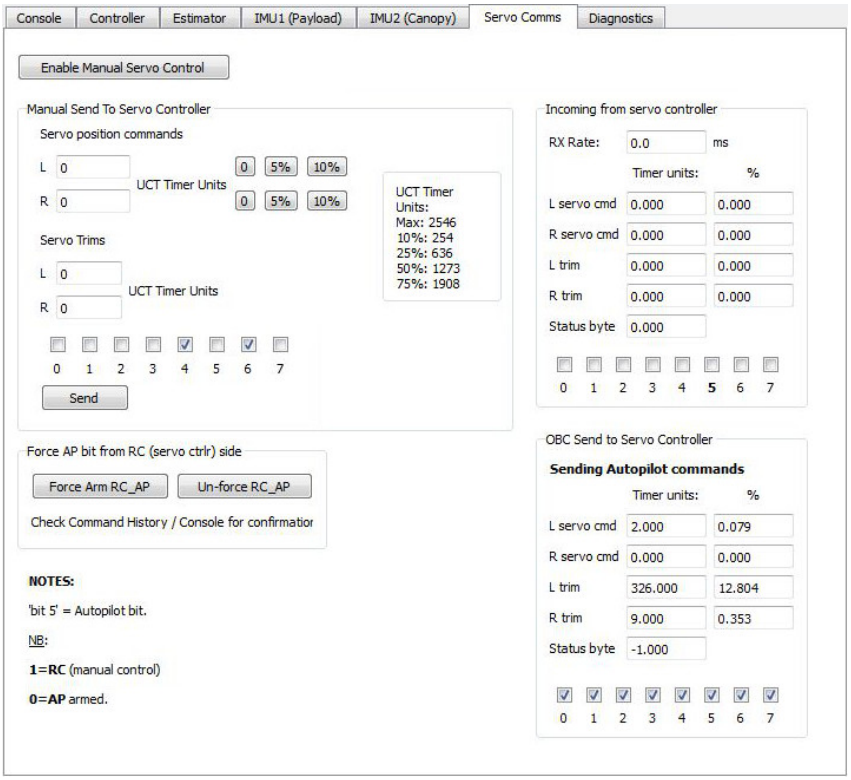


Figure 6.8 – GS servo comms tab.

6.4.5 Diagnostics Tab

The last tab is the *Diagnostics* tab, shown in Figure 6.9. This tab displays SD card information which can be used to troubleshoot SD card issues. On the right-hand side, a TCP/IP connection can be established by either hosting a server or becoming a client. This allows telemetry data to be streamed to a remote application over a TCP/IP network and allows the remote application to either listen as a server or function as a client. The TCP/IP streaming functionality was requested by the UCT team [17] in order to display current flight data for the SP, separate from the ESL GS. The package format is displayed in the GS tab for user reference and given below in Table 6.4.

6.4.6 Ground Station Data Logging

The GS logs all the received telemetry data, error messages and commands sent to the OBC to CSV files which are separated in folders for each GS session. MATLAB scripts are used to extract the logged data. Although the data is logged to the SD card at a higher rate, analysis of the GS data can help troubleshoot issues such as communication failures and enable quick flight data analysis on site at a flight test.

	Data	Unit	Data Length
Inertial Sensors:			
1	Payload Gyro X	°/s	4 bytes
2	Payload Gyro Y	°/s	4 bytes
3	Payload Gyro Z	°/s	4 bytes
4	Payload Acc X	m/s^2	4 bytes
5	Payload Acc Y	m/s^2	4 bytes
6	Payload Acc Z	m/s^2	4 bytes
7	Canopy Gyro X	°/s	4 bytes
8	Canopy Gyro Y	°/s	4 bytes
9	Canopy Gyro Z	°/s	4 bytes
10	Canopy Acc X	m/s^2	4 bytes
11	Canopy Acc Y	m/s^2	4 bytes
12	Canopy Acc Z	m/s^2	4 bytes
13	Magnetometer X	Gauss	4 bytes
14	Magnetometer Y	Gauss	4 bytes
15	Magnetometer Z	Gauss	4 bytes
GPS:			
16	GPS Latitude	°	4 bytes
17	GPS Longitude	°	4 bytes
18	GPS MSL	m	4 bytes
19	GPS Heading (Ground track)	°	4 bytes
20	GPS North Velocity	m/s	4 bytes
21	GPS East Velocity	m/s	4 bytes
22	GPS Down Velocity	m/s	4 bytes
State Estimator:			
23	EKF North Displacement	m	4 bytes
24	EKF East Displacement	m	4 bytes
25	EKF Down Displacement	m	4 bytes
26	EKF North velocity	m/s	4 bytes
27	EKF East velocity	m/s	4 bytes
28	EKF Down velocity	m/s	4 bytes
29	EKF Roll angle	°	4 bytes
30	EKF Pitch angle	°	4 bytes
31	EKF Yaw angle	°	4 bytes

Table 6.4 – GS TCP/IP package format.

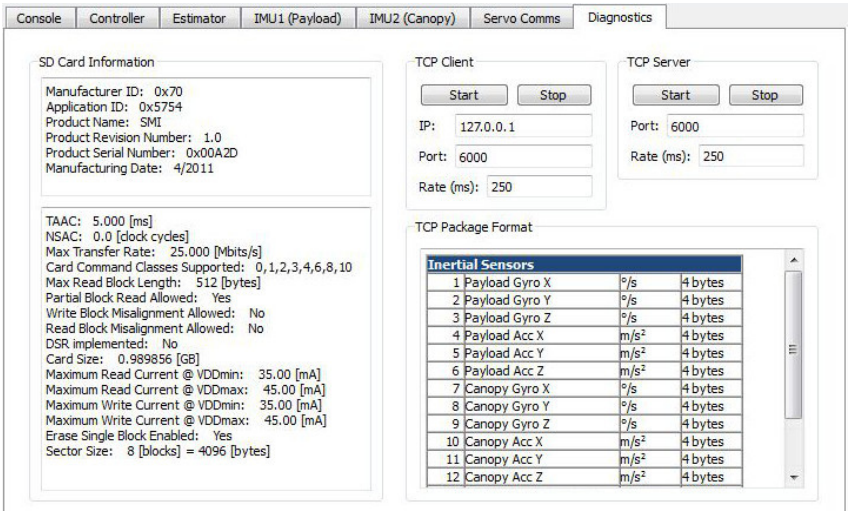


Figure 6.9 – GS diagnostics tab.

Chapter 7

Hardware In the Loop Simulation

In this chapter, the various flight control system (FCS) components will be tested in a hardware in the loop simulation (HILS), in order to verify the correctness of the FCS running on the actual hardware, assess the effects of sensor noise and other non-ideal conditions, and address implementation specific issues.

Firstly, each individual controller will be validated with the HILS separately. A controller is validated by ensuring that the output response of the controller running in the HIL simulation exactly matches the output response of the controller in the Simulink environment. Subsequently, the controller will be tested with simulated real-world imperfect conditions such as sensor noise.

The HIL test procedures are set up and carried out similar as would be done for actual flight tests in order to assess the procedures, gain confidence, understanding and experience in the entire process and fully test all the relevant components, including the ground station software, communications link and the on-board software.

An overview of the HIL framework is given in Section 7.1, after which the implementation of the FCS on the OBC is discussed in Section 7.2, followed by HIL simulations of the various FCS components in the subsequent sections.

7.1 HILS Overview

The HIL system enables the FCS to be tested on the actual on-board computer (OBC) by providing sensor and GPS measurements from the computer simulation and routing actuator commands to the simulation. The central component of the HIL simulation is the *HIL distribution board* which facilitates the connection and timing between the OBC and the simulation running in the Simulink environment. Figure 7.1 presents a block diagram of the HIL setup. The HIL distribution board connects to the OBC via the CAN bus (on which the OBC expects to receive sensor data and sends actuator commands) and a serial port through which GPS data is sent. When HIL is enabled, the OBC ignores actual GPS and sensor data and only accepts the simulated data from the HIL board. On the simulation side, the HIL board connects to a PC, using a serial port. A HIL Simulink block accepts sensor data from the parafoil simulation, sends the data to the HIL distribution board and receives actuator data from the HIL board which was generated by the OBC. Figure 7.2 shows the block diagram of the Simulink HIL simulation where the HIL block can be seen.

Referring to Figure 7.2, the *Aircraft Model* block contains the non-linear parafoil model. The model accepts brake deflection commands and a wind enable/disable signal as input and provides the parafoil position, velocity and attitude data is output. Sensor data is passed from the parafoil model to a *Sensor Model* block which packages the data appropriately

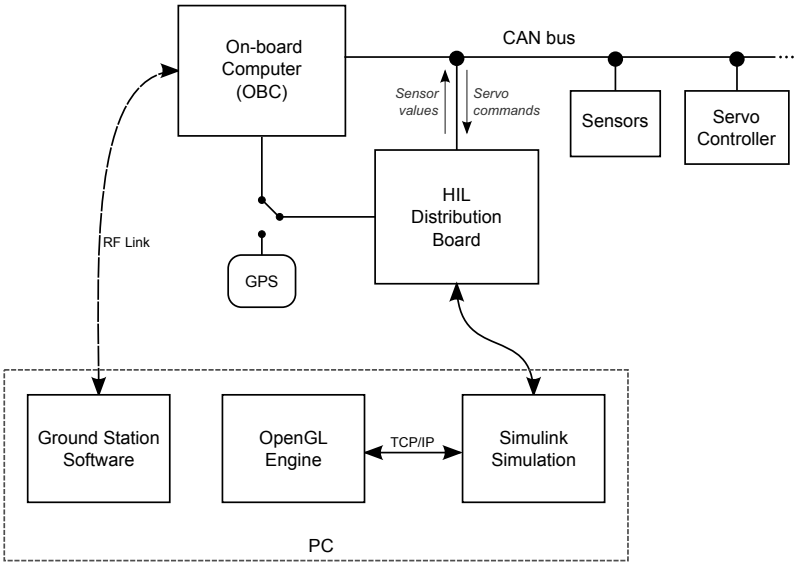


Figure 7.1 – Block diagram of HIL setup.

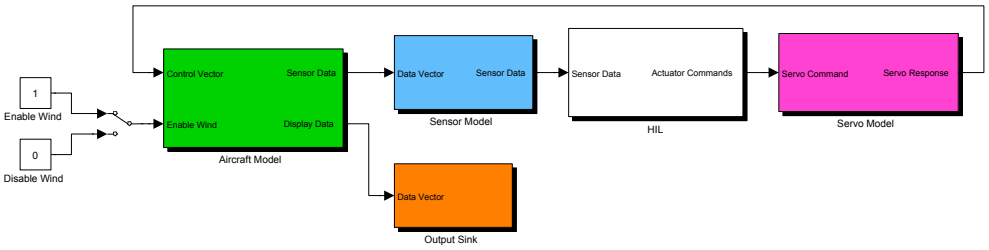


Figure 7.2 – Simulink HIL simulation.

to represent sensor measurements. The block also adds sensor noise and the final measurements are passed to the *HIL* block which handles the communication to and from the HIL distribution board and ensures that the simulation runs in real-time, synchronised with the HIL distribution board. On the right-hand side of Figure 7.2, the *Servo Model* block receives the actuator commands from the HIL block. The commands are passed through a filter representing the actuator dynamics and the final brake commands exit the block and are passed back to the aircraft model.

The remaining two items to be noted in Figure 7.1 are the *Ground Station Software* and the *OpenGL Engine*, running on a PC. The ground station software is used in the HILS process, as would be done for an actual flight test, to initialise and configure the OBC, enable the estimator and arm the autopilot at the appropriate time. The OpenGL engine software, developed by [31], receives the simulated parafoil position and attitude from the Simulink simulation (via the *Output Sink* block in Figure 7.2) and provides a 3D visualisation of the parafoil. A screenshot of the visualisation is given in Figure 7.3. The 3D parafoil and payload model were created by the UCT team [17].

It is evident from the presented HIL setup how the HIL simulation enables the OBC hardware and software and the ground station software to be tested under simulated flight conditions, while providing a useful graphical presentation of the parafoil.

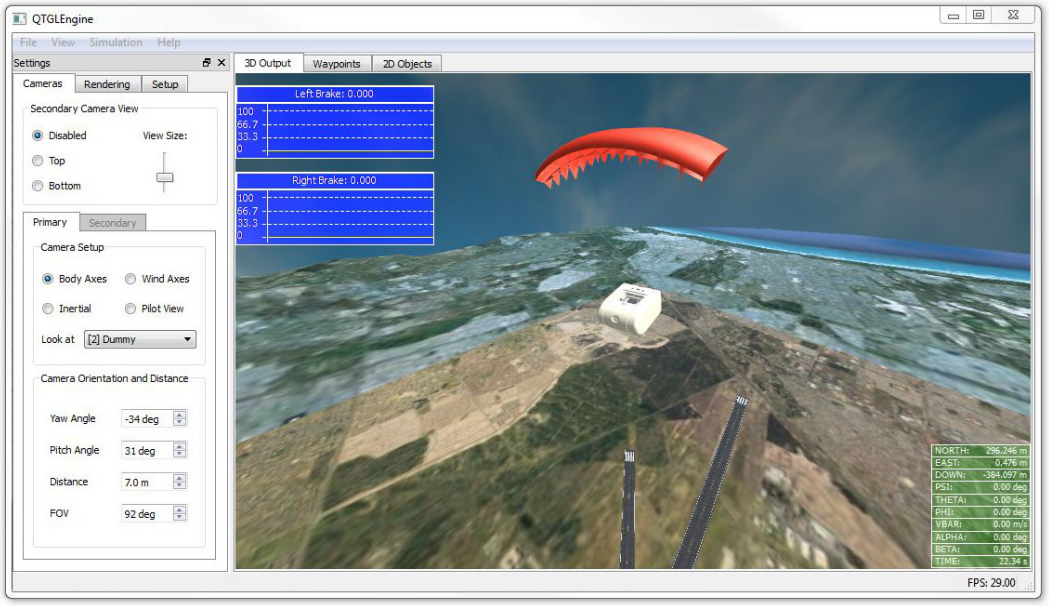


Figure 7.3 – GLEngine - 3D visualisation of the parafoil in flight.

The RMS values of the simulated sensor noise generated by the sensor model block are based on values from sensor data sheets and [27] and are listed in Table 7.1. In reality, the GPS position displays a random drift. However, this is not included in the simulation.

Sensor	Noise RMS
GPS velocity	0.5 m/s
GPS horizontal position	2.5 m
GPS altitude	4 m
Gyroscopes	0.4°/s
Accelerometers	0.1 m/s^2

Table 7.1 – Sensor noise levels for HILS.

The actuator model is approximated to the first order low pass filter in Equation (4.1.3) used for the FCS design. This imposes a bandwidth limit which is known to be well within the capability of the motors. Actuator commands sent to the simulation environment experience quantisation identical to that discussed in Section 6.3. In addition, the small latency expected from the servo communications between microcontroller B and the UCT microcontroller is also present in the HIL simulation due to a single cycle delay imposed by the HIL board. Other non-linearities of the parafoil motors are neglected and not modelled.

7.2 FCS Implementation on the OBC

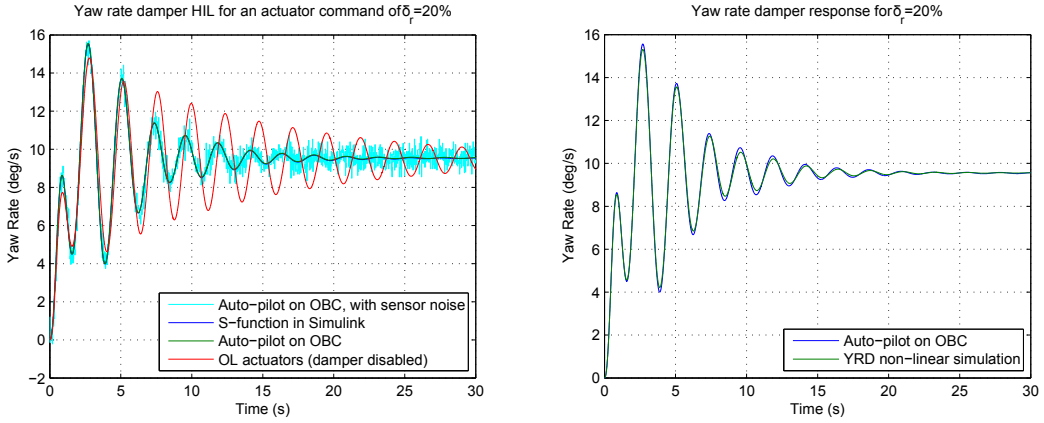
During the design process, the FCS components were implemented in the Simulink environment using the common tools and functions available. The final FCS implementation is in C on the OBC microcontroller. In preparation of this, the FCS was coded in C as a MATLAB Simulink add-in component, referred to as a s-function, in order to debug and verify the code alongside the original Simulink FCS implementation and the non-linear parafoil model. To

ease the code conversion process from the s-function to the OBC, the entire code structure, including variable and function names, data structures and procedures of the OBC were adopted for the s-function, enabling the source code files to simply be copied to the OBC source project with minimal interface adjustments. This greatly eases the debugging and code porting process. The microcontrollers perform floating point calculations with sufficient accuracy, allowing the data types to be directly ported between the two environments and allowing quantisation effects due to calculations to be neglected [27].

7.3 HILS of Inner-loop Controllers

This section discusses the HILS of the inner-loop controllers, which include the yaw rate damper, yaw rate controller and non-linear guidance controller. The HILS process is outlined for each test and is determined to be representative of the procedures that would be followed for an actual flight test. This allows the GS software to be tested alongside the FCS.

The yaw rate damper is tested by inducing the payload twist oscillation and observing the yaw rate with and without the damper enabled. The most straight forward method to induce the oscillation in a flight test is to apply an open loop actuator step input. The flight procedure is simple – the simulation is started, the yaw rate damper controller alone is enabled and the autopilot is armed, after which a constant actuator step command is applied. This is repeated with the yaw rate damper disabled for comparison.



(a) Yaw rate damper on OBC.

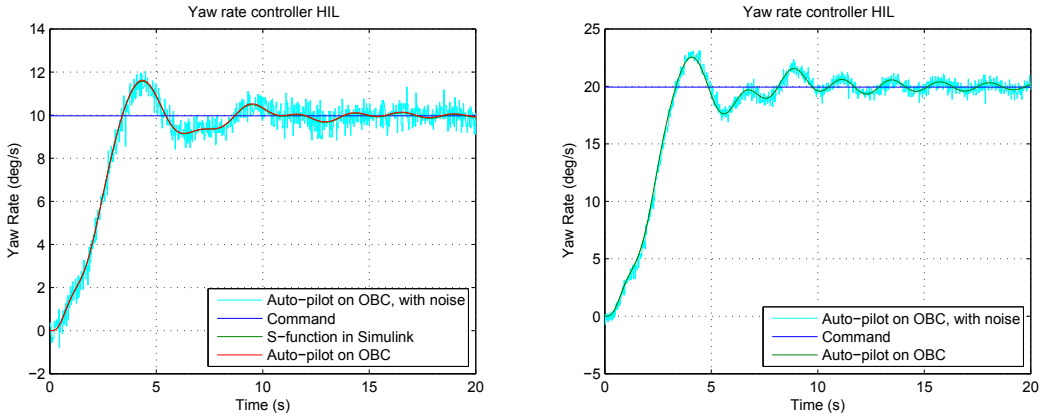
(b) OBC vs. Simulink.

Figure 7.4 – HIL simulation results for yaw rate damper.

Figure 7.4(a) displays results for the yaw rate damper HIL simulations for cases with and without the yaw rate damper enabled, as well as with gyroscope noise enabled. An open loop brake step of $\delta_r = 20\%$ was applied to induce the oscillation dynamics. The results of the damper running on the OBC and the one running in Simulink (using the same measurements) match exactly, verifying the correctness of the OBC code. The results also verify the working of the yaw rate damper under noisy conditions. In Section 4.2.1, the yaw rate damper step response for the linear and non-linear parafoil model were compared in the Simulink environment. The non-linear model response is now compared to the yaw rate damper running on the OBC in the HIL simulation as an additional test to verify that no errors were introduced when porting to the OBC. The comparison is given in Figure 7.4(b). A slight difference in the response can be observed due to the delay introduced by

the HIL simulation. Otherwise, the responses match very well, instilling confidence in the OBC implementation.

The yaw rate controller HIL simulation is performed similar to the previous, by providing a yaw rate step input command. Figures 7.5(a) and 7.5(b) show HIL simulation results for 10°/s and 20°/s yaw rate step commands, respectively. The response of the controller running on the OBC matches that of the controller running in Simulink with the same measurements and the controller performs well under noisy conditions. In Section 4.3.1, the



(a) 10°/s yaw rate step.

(b) 20°/s yaw rate step.

Figure 7.5 – HIL simulation results for yaw rate controller.

yaw rate controller design was verified with the non-linear parafoil model in the Simulink environment. The Simulink controller is now compared to the controller running on the OBC to further verify that no errors were introduced in the porting process. The comparison is given in Figure 7.6 for a 10°/s step command. Similar to the case for the yaw rate damper, only a small difference is visible in the responses, which is due to the delay introduced by the HIL simulation.

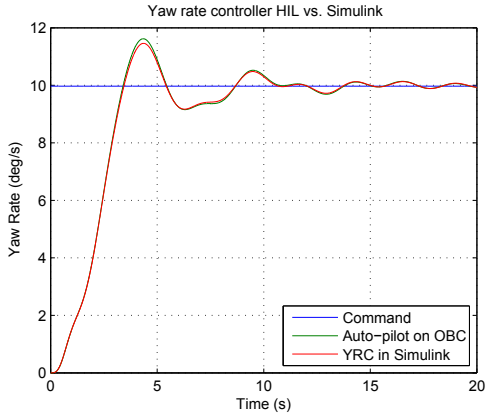


Figure 7.6 – Yaw rate controller HIL simulation results vs. Simulink.

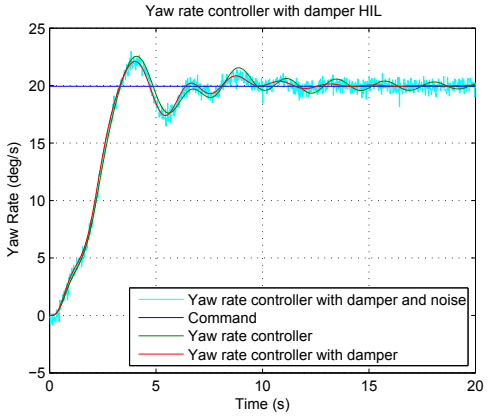


Figure 7.7 – HIL simulation results for yaw rate controller with yaw rate damper enabled.

Figure 7.7 shows HIL simulation results for yaw rate controller step responses with the yaw rate damper also enabled. Results indicate that the yaw rate damper does not degrade the yaw rate controller performance and damps out a small steady state oscillation.

The non-linear guidance method uses the horizontal velocity and position of the vehicle to determine the appropriate yaw rate command, which are provided by the on-board estimator. The HIL simulation of the non-linear guidance method will thus verify that both the controller and the estimator are functioning properly.

The first test entails tracking a straight line, in the direction of the initial vehicle heading, which is at a cross-track offset from the initial position. This is similar to providing a step input to the path follower. The test procedure for HIL and a flight test case is as follows:

- The OBC and simulation is started and the estimator is initialised and enabled.
- The non-linear guidance controller and the yaw rate controller are enabled.
- The step command magnitude is set to the desired cross-track offset.
- Once the estimator has converged and the vehicle is in steady state flight, the autopilot is armed and the step input command is issued. Since the path follower is enabled, the OBC interprets the step command by setting the desired trajectory to be a straight line in the forward direction, with a cross-track offset from the current position at a distance equal to the commanded step magnitude.

The HIL simulation results for the non-linear guidance controller, with $L_1 = 120$ m, a 50 m step and GPS and sensor noise added, compares well with the expected results from the controller simulations, as illustrated in Figure 7.8. For the case with noise, both the actual response and the noisy response as seen by the OBC are shown.

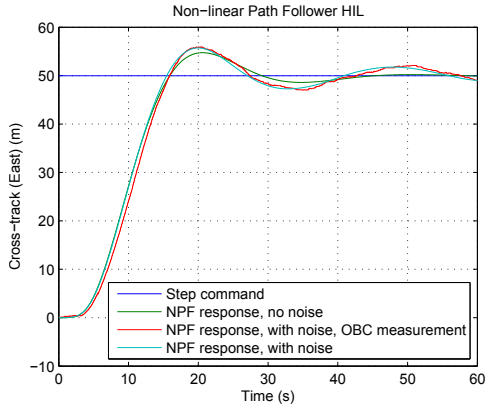


Figure 7.8 – HIL simulation results for non-linear guidance method.

7.4 Optimal TG HILS

The main objectives of the optimal TG HIL simulations are to:

- Identify and overcome implementation and other potential practical issues.
- Verify that the optimal trajectory planning and tracking procedures are functioning correctly.

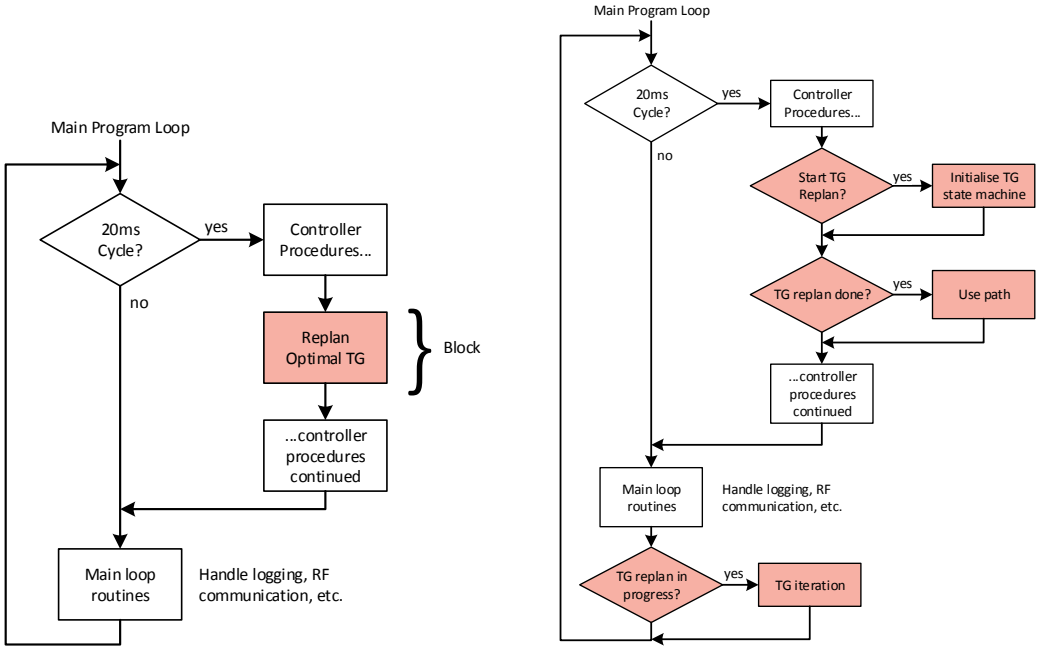
- Test the system in a windy and noisy environment and verify that the system behaves as expected.

7.4.1 Implementation Issues

When implementing the optimal TG algorithm on the OBC, a critical performance issue arose. While the algorithm executes fast enough on a PC for a trajectory with 20 or more nodes, execution on the OBC was too slow in the initial port, taking up to 7 seconds to complete. Additionally, problems were encountered due to insufficient memory. The algorithm procedure was optimised to increase processing speed and reduce memory usage. Three areas were identified where the algorithm could be optimised with regards to execution speed:

- Blocking vs. non-blocking execution
- Number of nodes used in the optimal trajectory
- Optimal trajectory search algorithm

For the first point, consider the flow diagrams representing the implementation of the optimal TG algorithm on the OBC in Figure 7.9 for the initial and revised implementations, with highlighted blocks related to the optimal TG routines. Figure 7.9(a) depicts the initial



(a) Initial (blocking) implementation.

(b) Revised (non-blocking) implementation.

Figure 7.9 – Flow diagrams of the optimal TG implementation on the OBC.

implementation. The control system (FCS) procedures are run at 20 ms intervals. When it is required that an optimal TG path be calculated, the optimal TG algorithm is run, blocking the other controller procedures and the main program loop until it is done, as indicated in the flow diagram. If the algorithm takes long to execute, this presents a problem to the system as critical procedures can not be performed within the required 20 ms interval. This problem

was alleviated by reimplementing the optimal TG procedure as state machines, allowing the execution of the algorithm to be split over several main program loop iterations such that a single, short, task is performed for every iteration, not blocking the rest of the system from executing. The revised implementation is depicted in Figure 7.9(b). In the control system procedures, the optimal TG state machine is flagged to start if an optimal TG replan is necessary. The rest of the control procedures are allowed to be executed while the algorithm is completed in the "background."

Regarding the second point listed above, reducing the number of nodes used for the optimal trajectory to 6 gives an acceptable trajectory resolution while requiring less execution time on the OBC.

Thirdly, the search algorithm used to find the optimal trajectory was modified to be more efficient. In Section 5.2.4 it was noted that a simple linear search was implemented to find the optimal algorithm parameter τ_f. The search was modified to use larger intervals when iterating over the range of τ_f values. When finished, the search is refined around the minimum sampled value. This drastically reduces the number of algorithm executions required while remaining simple and robust as a search technique.

7.4.2 HIL Simulation

The HIL flight test procedure for optimal TG is as follows:

- The GS software is used to set up the controller parameters and the flight plan. The landing target is chosen far enough from the simulation start location to allow some time for the estimator to be initialised once the simulation has been started.
- The simulation is started and the estimator is initialised. The vehicle travels in a straight line. When deemed adequate and before the starting point of the TG phase is reached, the autopilot is armed.
- The vehicle eventually reaches the TG starting point, and the optimal TG procedure is carried out.

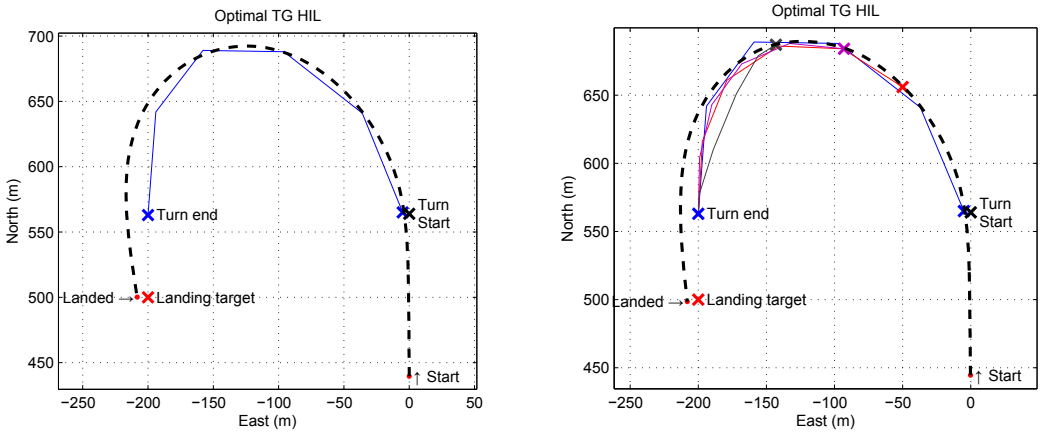
Initial tests ensure that the planning and tracking procedures are functioning correctly. Table 7.2 lists the user selectable parameters used for the HIL simulation.

Parameter	Value
Non-linear guidance L_1	120 m
Optimal TG replan frequency	4 s
Ideal TG circle radius	100 m
Preferred initial conditions for TG	Excess altitude of 20 m
FA	4 s
Landing target (north,east)	500, 200 m

Table 7.2 – User selectable parameters for optimal TG HIL simulation.

Figure 7.10 displays HIL results for optimal TG procedures performed under favourable conditions (i.e. no wind disturbances). The TG turn start was reached with an excess height of 18 m. Figure 7.10(a) illustrates the case for which replanning was disabled and Figure 7.10(b) the case for which it was enabled. Since no wind is present and the vehicle did not notably deviate from the desired path, the results are similar for both cases.

The next step is to run the simulation under windy conditions. Wind is generated by filtering white noise to produce wind gusts with a RMS of 0.6 m/s that last approximately a second.



(a) Optimal TG HIL.

(b) Optimal TG HIL with replanning.

Figure 7.10 – Optimal TG HIL simulations under good conditions.

The flight test procedure is similar to the previous flights. Initially, the flight is started with no wind. When the vehicle is near the ideal TG start point, the wind input is enabled and the optimal TG procedure is observed. Results are plotted in Figure 7.11. The flights with replanning disabled and enabled are depicted in Figures 7.11(a) and 7.11(b), respectively, with the random generated wind gusts shown in Figure 7.11(c). While the HIL simulations are run with a limited set of generated wind gusts, the results verify the functioning of the optimal TG algorithm and trajectory tracking on the OBC and instil confidence in the implementation.

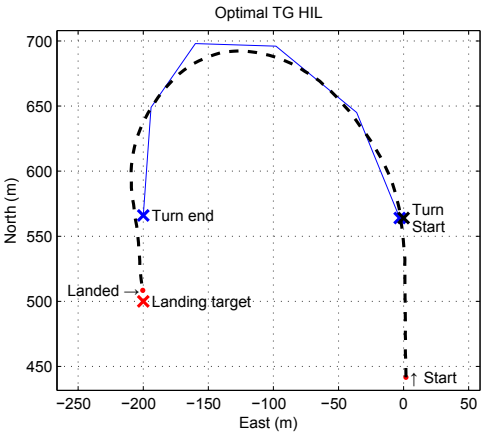
7.5 Full Flight HILS

Moving on to the full flight HIL simulation, the objectives are to test the entire process from setting up the flight plan with the GS software to verifying that the FCS and estimator on-board the OBC function as expected. The flight plan is as follows:

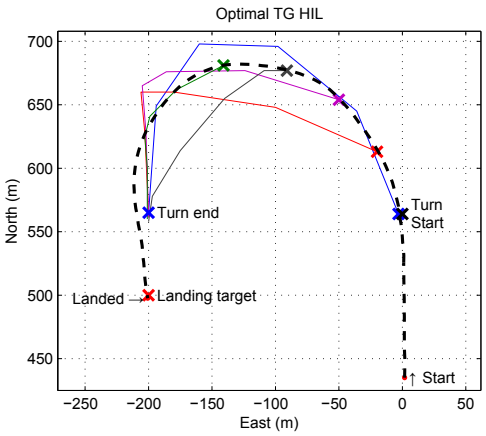
- With aid of the GS software, the appropriate controllers are armed and configured. The distance from the start location to the first waypoint (start of the homing phase) is chosen long enough to enable the estimator to converge, the vehicle to settle into steady state flight and the user to perform last-moment checks before arming the autopilot. Arbitrary waypoints are added and the flight plan is reviewed, before confirming and switching to flight mode.
- The simulation is started and the estimator is enabled. As the vehicle nears the first waypoint, the autopilot is armed.
- The user can monitor the vehicle status with the GS and observe the flight on the graphical display.

The preflight path planning settings are listed in Table 7.3. A single intermediate waypoint was added. Wind gusts and sensor noise is also enabled.

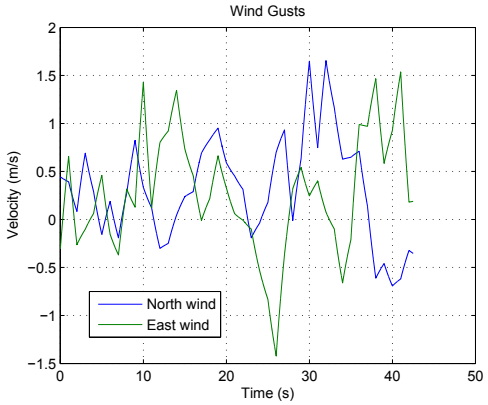
The full flight HIL result is shown in Figure 7.12. Starting at point (0,0), the preflight planned trajectory is shown in blue, connecting the three waypoints (*wp0*, *wp1* and *wp2*). The preflight EM circle, indicated in blue, is calculated with radius 132 m and 2 revolutions are required. The actual flown trajectory is indicated by the thick black dashed line. The



(a) Optimal TG replanning not enabled.



(b) Optimal TG replanning enabled.



(c) Wind gusts used for HIL simulation.

Figure 7.11 – Optimal TG HIL simulations with wind and sensor noise disturbances.

Parameter	Value
Initial position (north, east, down)	0, 0, -1600 m
Pre-homing distance	600 m
Landing target (north, east)	700, 600 m
Optimal TG replan frequency	4 s
Ideal TG circle radius	100 m
FA duration	4 s
Intermediate waypoint	350, 300 m
Number of EM circles and radius	$n = 2$, $R = 132$ m

Table 7.3 – Preflight path planning settings for HIL simulation.

effect of the wind gusts is evident from the imperfect tracking of the initial straight line leading to the first waypoint. The Dubins trajectory is followed and the homing phase is completed at *wp2*, marking the start of the EM phase. At this point, the first EM circle radius is recalculated based on the current height and velocity, resulting in a smaller circle than the preflight case, with a radius of 116 m. When the first EM circle is completed, the second EM circle radius is calculated similar to the first, resulting in a slightly larger circle than the preflight planned circle, with a radius of 141 m. The EM phase is eventually completed when the vehicle reaches *wp2* again and the straight leg leading to the start of the TG turn is followed. When the TG start criteria is met, the optimal TG phase starts and several trajectories are planned as the vehicle progresses through the turn, one every 4 seconds. The landing target is eventually reached and the parafoil lands 7 m away from the desired target.

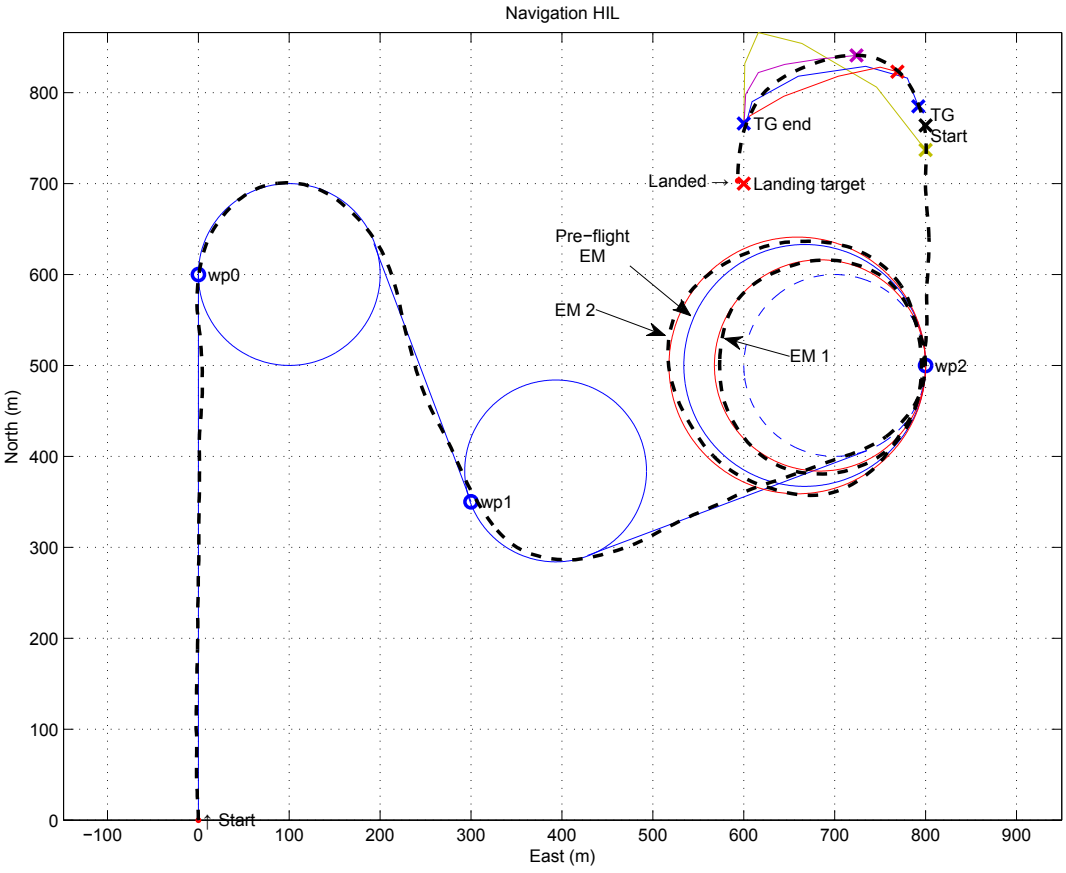


Figure 7.12 – HIL simulation results for complete flight, with sensor noise and wind disturbances.

7.6 Summary and Conclusion

This chapter presented an overview of the HIL framework and simulation process, after which HIL simulations for the various FCS components were presented and discussed. The simulations verified that the components function correctly on the actual hardware and discussed some implementation issues involved in porting from the MATLAB simulation

environment to the actual hardware. The most notable implementation issue encountered is that of the optimal TG algorithm, due to the limited processing power of the OBC microcontroller. This led to a compromise of reducing the number of nodes used in the optimal TG trajectory.

A full flight mission HIL simulation was run, testing the FCS under simulated wind gusts and sensor noise. The simulation results verify the functionality of the FCS on the OBC hardware, as well as the ground station software and the on-board estimator. Together, the non-linear parafoil model and the HIL simulation are shown to be a valuable tool for testing and preparing the system to be integrated in the actual parafoil and used for practical flight tests. Finally, the simulations also show the applicability of the linearised model which was used in the FCS design process.

Chapter 8

Flight Tests

Several flight tests were conducted with the UCT parafoil test system throughout the span of this project. All of the tests are not directly related to the work developed in this project but illustrate the evolution of the entire parafoil system and are noted here to provide a sense of the broader development scheme of which this project is a part of. The flight tests tested the following main capabilities, listed in chronological order:

- Launcher system
- Remote-control (RC)
- Autopilot (AP)

The launcher system tests were performed as part of the development of the UCT launcher system and were concluded in the first half of 2012. The tests were carried out by the UCT team at the UCT campus and a decommissioned quarry in Durbanville and involved launching the parafoil with a dummy weighted payload. The launcher design, configuration, parafoil rigging settings and the launch procedure were evaluated and at the conclusion of the tests the launching mechanism and procedure was deemed reliable to progress to RC flight tests.

RC tests commenced in August 2012. For these tests, the steering unit, including the servo motors and the UCT microcontroller, and the avionics unit were integrated into the payload box. Through these tests a pilot controllable RC vehicle with data logging capability was established.

One set of AP flight tests were conducted in April 2013 in which the parafoil was controlled autonomously by the FCS for the first time. Although only a small number of AP tests were conducted, the tests marked the successful integration of autonomous control with the actual system, establishing the ground work for future autonomous control flight tests.

The following sections outline some of the flight tests for which data was logged by the avionics unit. The testing was performed at a decommissioned quarry in Durbanville for which permission to test at was obtained from the owner. The testing location contains cliffs with approximately 20 m high steps forming a semi-circle amphitheatre around a relatively flat base. The cliff steps allow various accessible levels of different heights from which the parafoil can be launched to land in the flat base area. The height available for the flight tests is very limited, resulting in an average flight time of approximately 20 s. Figure 8.1 shows the parafoil after being launched in one of the flight tests.

Figure 8.1 – The parafoil in flight after being launched at the quarry.

8.1 RC Flight Tests

22 August 2012

The goal of the flight tests were to test the manual RC of the parafoil. On this day, the parafoil was launched from the second cliff level, approximately 40 m above the base. One successful flight was performed. However, steering issues were experienced. The parafoil launched successfully but was pushed in the direction of a lake adjacent to the landing area by wind. Attempts by the RC pilot to gradually guide the parafoil away from the lake failed to obtain a response from the parafoil and the pilot was forced to command a full left deflection at the last moment, resulting in a hard ground impact but ensuring the safety of the vehicle. Post-flight analysis indicated that the steering lines had not been set up correctly.

16 November 2012

RC issues were experienced again which resulted in a severe crash into a cliff face. The payload encasing was damaged as well as one of the motor gearboxes. Fortunately, the gearbox damage was not irreparable. Considering the immense impact, the system survived extremely well. Apart from some components coming loose and shifting, the avionics was not affected.

6 March 2013

Several successful flight tests were performed in which the RC piloting was systematically tested across the flights using simple turning and flare manoeuvres. These tests verified the functioning of the RC system and confidence was gained in the launching, rigging and manual controllability of the system.

8.2 Autopilot Tests on 25 April 2013

The first flight of the day was an uncontrolled flight to verify that the launcher and rigging setup was working correctly, as well as the AP switch and the RC. In the second flight, all the FCS controllers were disabled and a 30% right brake deflection step was commanded in order to determine the $\dot{\psi}$ to δ_a gain, K_δ. The 30% deflection caused an approximate 20°/s turn resulting in $K_\delta = \frac{20}{30} = 1.5$. This value was used for the subsequent flights. Flights 3 and 6 tested the yaw rate controller and heading controller, respectively, while flights 4 and 5 were unsuccessful due to technical and rigging issues.

Flight 3 - Yaw rate controller step input

With $K_\delta = 1.5$ as determined from the previous flight, the AP was commanded to track a 20°/s yaw rate step input. A turn was observed which started gently followed by a sudden sharp increase. The logged data and video footage revealed a large increasing yaw oscillation with a frequency between 4.2 and 5.2 rad/s when the yaw rate controller step was engaged. Figure 8.2 gives the yaw rate controller input command, yaw rate measurement and brake deflection for the flight. The AP was armed after 2 s, indicated by the first non-zero value of δ_a. The step input command was initiated just after 4 s, after which a very prominent, growing oscillation is seen. The clipping of the δ_a signal is due to a 50% limit imposed on the δ_a command. The AP was disabled just after 9 s, indicated by δ_a returning to zero with a steady slope. The yaw rate controller in use was the initial PI and notch compensation design of Equation (4.3.9) which rejects frequencies around 2.6 rad/s but fails to reject higher frequencies. The payload relative twist oscillation of the parafoil seems to be at a different frequency from that of the linear model. In the light of this, the yaw rate compensation design was amended, as discussed in Section 4.3.

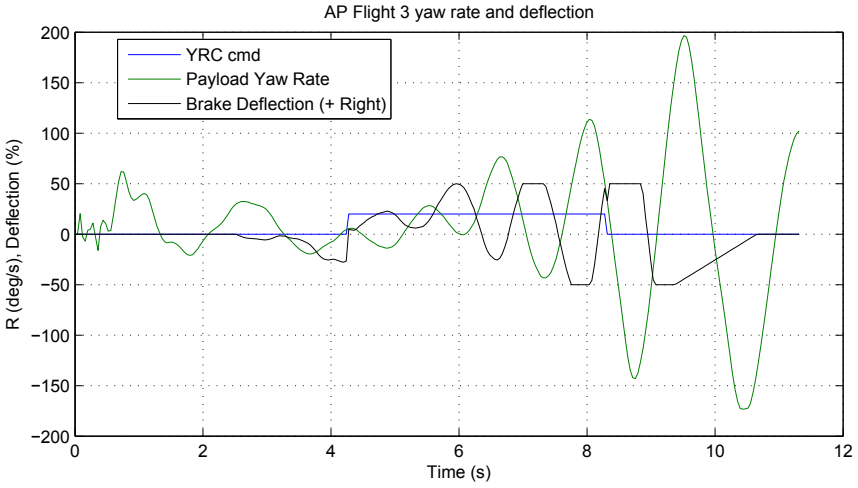
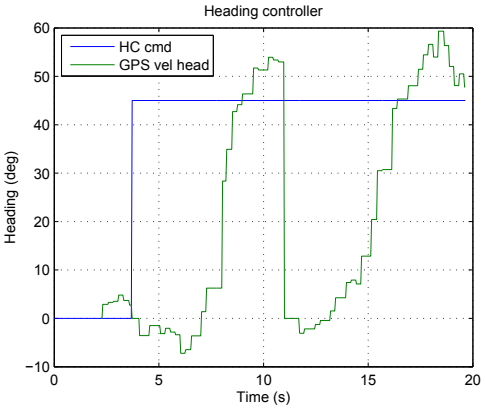


Figure 8.2 – AP flight test 3 results – yaw rate controller.

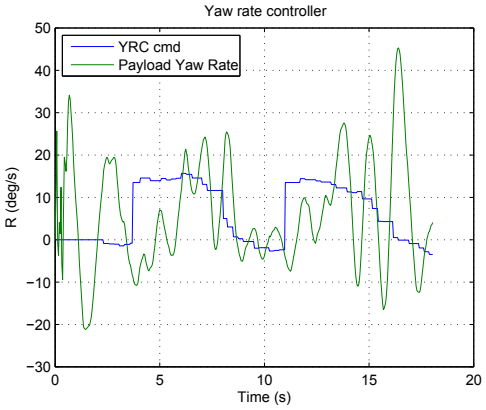
Flight 6 - Heading controller step input

For this test, step input commands were provided to the heading controller. Figure 8.3 displays the heading controller response, yaw rate controller response and the flown track. The low frequency heading measurement in Figure 8.3(a) is the GPS velocity vector heading that was used for the heading measurement due to problems experienced with the magnetometer and the estimator on the day of testing. Figure 8.3(a) indicates that the heading

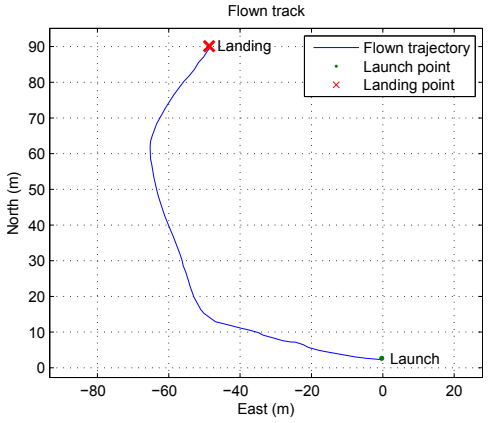
step command was initiated at approximately 4 s. A second step command was issued at approximately 11 s, indicated by the sharp fall of the heading measurement to zero, since the shown heading is relative to the heading when the command was issued. The heading controller response is quick but experiences large overshoot due to the low frequency measurement used and the inadequate yaw rate controller. Increasing yaw rate oscillations are present in the yaw rate controller response in Figure 8.3(b), similar to the previous flight test.



(a) Heading controller response.



(b) Yaw rate controller response.



(c) Flown track.

Figure 8.3 – AP flight test 6 results – heading controller.

8.3 Conclusions

Although only a small portion of the flight tests that were conducted during the course of this project directly relate to the work developed in this project, all of the tests illustrate the development towards a complete autonomous parafoil vehicle and testing facility and greatly contributed to the integration between the work developed in this project and the actual system.

The AP tests exposed a serious design flaw in the initial yaw rate controller, which was subsequently amended. This also raises the concern that attention should be given to

identifying the exact characteristics of the actual system and reconciling with the computer simulation.

Chapter 9

Summary and Recommendations

9.1 Summary

The design and implementation of a FCS for an autonomous parafoil was presented in this thesis, capable of planning and following a flight path from an arbitrary starting position to a landing target, compensating for external disturbances to land as close as possible to the landing target.

The FCS design was based on a linear model, obtained by numerically linearising a non-linear parafoil model. Analysis of the linear model indicated a stable but poorly damped payload twist oscillation for which a stability augmentation design was proposed. Successive control loops and a guidance controller were designed, enabling autonomous path following. Finally, mission control and path planning methods were created, using EM and optimal TG strategies to compensate for external disturbances.

The FCS was implemented on existing avionics hardware and verified through HIL simulations. The system was then integrated into an existing parafoil system.

The project met the goals of designing a FCS enabling autonomous flight, implementing it on existing hardware and integrating with the existing parafoil system. The design and hardware was thoroughly tested in simulation but limited practical testing was performed due to project time constraints.

Apart from the designed FCS, developments by the current project which are of valuable use to the UCT parafoil project and future autonomous parafoil research at the ESL are listed below:

- The non-linear MATLAB simulation of [16] was originally based on a different parafoil and payload system. With assistance from C. Redelinghuys, the simulation was adapted to the UCT system.
- The non-linear parafoil simulation of [16] was ported to a C s-function in Simulink from the C++ implementation of A. Grunwald [17].
- The integration with the UCT system, including the servo communications between the avionics and the current UCT microcontroller. This is not only a fundamental requirement for autonomously controlling the parafoil, but enables the RC commands to be logged in the avionics unit.
- The application created to test the servo communication between microcontroller B and the UCT microcontroller, described in Section 6.3, is an invaluable testing and troubleshooting tool which could aid future development.

- As previously mentioned, the ESL avionics and GS system had already been adopted by the UCT team for data logging purposes. However, the components were based on relatively old versions of the OBC and GS software. The system was upgraded to the latest version of the OBC and GS software and adapted to specifically suit the parafoil.
- The avionics, GS software, HIL simulation environment and the openGL 3D display was previously only configured for fixed wing and rotary wing UAVs. The systems were modified and configured to suit the needs of this project, thus adding compatibility with the parafoil.
- Functionality was incorporated into the MATLAB non-linear parafoil simulation to automatically linearise the model and initialise the simulation with a user defined eigenvector in order to excite a mode of motion.
- A simple GUI front-end to the original MATLAB non-linear parafoil simulation which alleviates the user from editing various script files to set simulation parameters. This provides a convenient way of easily and quickly running a simulation without knowledge of the underlying file structure.

9.2 Recommendations

Some recommendations for additions and research which would valuably attribute to the project are discussed below.

System identification to determine more accurate vehicle model

The yaw rate damper and yaw rate controller designs are based on a specific parafoil configuration. Reconciling the computer model with the actual vehicle to be used will enable more accurate simulation and more robust design, especially with regard to the payload relative twist motion frequency. Additionally, some form of feedback from the servo controller would be valuable in determining the exact response and delays of the brake actuation. Currently, the servos are controlled open loop by the UCT microcontroller and no servo position information is available.

Less aggressive yaw rate control

The yaw rate controller was designed with the aim of maximising response speed. The applicability of the controller still needs to be evaluated with flight tests. An alternate strategy is to approach the design by not aiming for a quick response, but ensuring slow, smooth and gradual motion. This could result in a simpler, more robust design applicable for a wider range of vehicles. In addition, less aggressive control could result in longer battery life. Another strategy is to use a hybrid of the control methods – less aggressive control for homing and EM phases and more aggressive and accurate control for the path critical phases such as TG. A thorough study on the comparison of the approaches could be useful.

Flare

The flare manoeuvre was not considered in the scope of the current project but is valuable to provide a softer landing. Timing is paramount to the execution of the flare and requires further research and potentially additional sensing equipment in order to accurately detect the ground. While the current system has proven to be able to withstand the hard impacts of rough landings and even severe crashes, implementing a well controlled flare manoeuvre

could extend the life expectancy of the equipment and contribute to eventually using the vehicle to deliver more delicate payloads.

Real-time wind estimation

The optimal TG method is designed to be able to take a constant wind, parallel to the direction of landing, into account. The current work described a simple wind estimation technique in Section 5.4.1 but also illustrated the limits of the technique in that it requires excessive altitude. As a recommendation for future work, a real-time wind estimation technique to obtain the mean steady state wind could be useful for the optimal TG phase. Taking the wind into account during the homing and EM phases would further add great value to the path planning and overall mission control.

Upgrade computational abilities

A faster microcontroller is needed to process the optimal TG algorithm faster, with more nodes. Faster systems are already in use at the ESL and have been implemented in recent projects. Alternatively, the computation scheme on the current OBC can be revised and optimised. A scheme where the second microcontroller offloads some of the processing can be used, for example, to enable faster trajectory calculations.

More accurate kinematic model for optimal TG algorithm

In the optimal TG trajectory calculation, a kinematic model that more closely represents the actual parafoil can be implemented to provide more realistic trajectories. This will, however, be more processor intense and require a faster microcontroller, as noted above.

Extend optimal trajectory planning to homing and EM phases

The optimal TG algorithm can be applied to other stages of the flight such as homing and EM phases to create smooth, natural trajectories which could be adapted en route based on the current vehicle position similar to the optimal TG phase.

Investigate rigging changes and aerodynamic damping for payload twist oscillation

It was indicated through comparison of simulations that aerodynamic damping may play a role in the damping of the payload relative twist oscillation. Additionally, some sources claim that rigging could play a role in the damping of the mode. Investigating these effects could potentially lead to a system with a better damped payload yaw oscillation with relatively inexpensive modification to the payload or rigging setup.

Replace UCT microcontroller with ESL servo board and consolidate system architecture

The current UCT microcontroller is a relatively unpowerful processor with limited functionality. A *servo board* was designed by [28] and has been used extensively in several successful UAV projects at the ESL. The board incorporates seamlessly with the ESL avionics on the CAN bus with several features, including:

- RS232 port to which the UCT RF modem can be directly connected.
- Alternatively, 8 PWM inputs are available which is used to receive signals from a remote control unit.

- Two pairs of input pins can monitor battery voltages.
- 16 output pins, usually used as PWM output to drive servos. However, some of the pins can be reconfigured as general I/O and used for other tasks.

Replacing the UCT microcontroller with the servo board could provide a better interface to the servo controller and allow servo position information to be received. In addition, microcontroller B would be free and can be used to offload some of the processor intensive tasks from microcontroller A.

Appendix A

Simulation Parameters

The various parameters that define the parafoil used in the non-linear simulation are listed below. All coordinates, unless otherwise stated, are given in parafoil geometric axes, which are parallel to the parafoil aerodynamic axes and with origin at the front tip of the wing, centred laterally, on the bottom chord of the canopy. In this case, the aerodynamic and geometric axes origins coincide.

Property	Value
Canopy wing chord	1.56 m
Canopy projected wing span	3.9 m
Canopy aerodynamic reference area	6.07 m^2
Canopy mass (including lines)	1.5 kg
Canopy trapped air volume	0.608 m^3
I_{pxx}	2.83 kgm^2
I_{pyy}	0.45 kgm^2
I_{pzz}	3.27 kgm^2
I_{pxz}	0 kgm^2
Centre of mass coordinates of canopy, trapped air volume and lines (x, z)	(-0.78, 0)
Centre of mass of canopy and lines only (x, z)	(-0.78, 0)
Centre of mass of air volume of canopy (x, z)	(-0.78, 0)
Hinge coordinates (x, z)	(0.39, 4)
Payload mass	25.6 kg
I_{Lxx}	1.08 kgm^2
I_{Lyy}	1.39 kgm^2
I_{Lzz}	1.84 kgm^2
I_{Lxz}	0 kgm^2
Payload aerodynamic reference area	0.229 m^2
Payload aerodynamic reference length	0.71 m
Payload centre of mass behind leading edge	0.4 m
Hinge coordinates in payload axes (x, z)	(-0.01, -0.906)
Starboard hinge y coordinate in payload axes	0.245

Table A.1 – Parafoil and payload physical properties.

The parafoil canopy aerodynamic coefficients for zero deflection are given in Table A.2.

Redelinghuys [16] obtains aerodynamic coefficients by manipulating results from a commercially available CFD code. A fitting process is used, resulting in a matrix of values, associated

Coefficient	Value
C_{D0}	0.0980
C_{Y0}	0
C_{L0}	0.3386
C_{l0}	0
C_{m0}	-0.1209
C_{n0}	0
$C_{Y\beta 0}$	-0.0053
$C_{l\beta 0}$	0.0077
$C_{n\beta 0}$	0.0021
C_{Dp0}	-0.0370
C_{Yp0}	0.5111
C_{lp0}	-0.7101
C_{np0}	-0.1809
C_{Dq0}	0.0665
C_{Lq0}	2.9013
C_{mq0}	-1.1396
C_{Yr0}	0.0867
C_{lr0}	-0.1314
C_{nr0}	-0.0498

Table A.2 – Parafoil canopy aerodynamic coefficients at zero brake deflection.

with a particular brake setting, relating angle of attack and relative yaw to the desired coefficients. Matrices are given below, for 5 different brake settings ($0\%, 25\%, 50\%, 75\%$ and 100%). Using the equations in [16], the aerodynamic coefficients can be determined for a particular angle of attack, relative yaw and brake setting.

$$\mathbf{M}_0 = \begin{bmatrix}
8.82\times10^{-2} & 1.56\times10^{-3} & 3.29\times10^{-4} & 0 & 0 & 0 & 0 \\
-0 & -0 & -0 & -5.84\times10^{-4} & 3.55\times10^{-8} & -7.63\times10^{-6} & 2.63\times10^{-7} \\
1.16\times10^{-1} & 6.23\times10^{-2} & 0 & 0 & 0 & 0 & 0 \\
-0 & -0 & -0 & 7.61\times10^{-4} & -3.14\times10^{-8} & -5.13\times10^{-6} & -2.17\times10^{-7} \\
-64\times10^{-2} & -1.71\times10^{-2} & 5.28\times10^{-5} & 0 & 0 & 0 & 0 \\
-0 & -0 & -0 & 8.65\times10^{-5} & -6.66\times10^{-9} & 2.2\times10^{-6} & -35\times10^{-7} \\
-5.87\times10^{-3} & 1.46\times10^{-4} & 6.41\times10^{-6} & 0 & 0 & 0 & 0 \\
8.48\times10^{-3} & -2.12\times10^{-4} & -15\times10^{-6} & 0 & 0 & 0 & 0 \\
1.42\times10^{-3} & 1.94\times10^{-4} & -6.68\times10^{-7} & 0 & 0 & 0 & 0 \\
-3.7\times10^{-2} & 0 & 0 & 0 & 0 & 0 & 0 \\
4.87\times10^{-1} & 6.95\times10^{-3} & -8.42\times10^{-5} & 0 & 0 & 0 & 0 \\
-6.94\times10^{-1} & -4.92\times10^{-3} & 19\times10^{-4} & 0 & 0 & 0 & 0 \\
-16\times10^{-1} & -2.1\times10^{-2} & 4.65\times10^{-5} & 0 & 0 & 0 & 0 \\
3.5\times10^{-3} & 1.73\times10^{-2} & 7.48\times10^{-5} & 0 & 0 & 0 & 0 \\
2.87 & 9.79\times10^{-3} & -3.16\times10^{-4} & 0 & 0 & 0 & 0 \\
-1.12 & -6.85\times10^{-3} & 1.74\times10^{-4} & 0 & 0 & 0 & 0 \\
1.61\times10^{-1} & -28\times10^{-2} & -1.54\times10^{-5} & 0 & 0 & 0 & 0 \\
-2.39\times10^{-1} & 3\times10^{-2} & 8.52\times10^{-6} & 0 & 0 & 0 & 0 \\
-6.13\times10^{-2} & 35\times10^{-3} & 4.7\times10^{-5} & 0 & 0 & 0 & 0
\end{bmatrix} \tag{A.0.1}$$

$$\mathbf{M}_{25} = \begin{bmatrix}
1\times10^{-1} & 1.74\times10^{-3} & 3.2\times10^{-4} & 0 & 0 & 0 & 0 \\
8.75\times10^{-3} & -7.28\times10^{-5} & -13\times10^{-5} & -5.84\times10^{-4} & 3.55\times10^{-8} & -7.63\times10^{-6} & 2.63\times10^{-7} \\
1.23\times10^{-1} & 6.18\times10^{-2} & 0 & 0 & 0 & 0 & 0 \\
-8.8\times10^{-3} & 1.31\times10^{-4} & 1.43\times10^{-5} & 7.61\times10^{-4} & -3.14\times10^{-8} & -5.13\times10^{-6} & -2.17\times10^{-7} \\
-7.15\times10^{-2} & -1.7\times10^{-2} & 6.46\times10^{-5} & 0 & 0 & 0 & 0 \\
45\times10^{-3} & 6.1\times10^{-5} & 7.12\times10^{-6} & 8.65\times10^{-5} & -6.66\times10^{-9} & 2.2\times10^{-6} & -35\times10^{-7} \\
-5.65\times10^{-3} & 1.44\times10^{-4} & 6.29\times10^{-6} & 0 & 0 & 0 & 0 \\
9.28\times10^{-3} & -2.82\times10^{-4} & -5.47\times10^{-7} & 0 & 0 & 0 & 0 \\
1.41\times10^{-3} & 1.99\times10^{-4} & -7.16\times10^{-7} & 0 & 0 & 0 & 0 \\
-3.4\times10^{-2} & 0 & 0 & 0 & 0 & 0 & 0 \\
4.85\times10^{-1} & 6.68\times10^{-3} & -8.31\times10^{-5} & 0 & 0 & 0 & 0 \\
-6.91\times10^{-1} & -4.48\times10^{-3} & 18\times10^{-4} & 0 & 0 & 0 & 0 \\
-14\times10^{-1} & -25\times10^{-2} & 4.56\times10^{-5} & 0 & 0 & 0 & 0 \\
1.58\times10^{-2} & 1.76\times10^{-2} & 5.29\times10^{-5} & 0 & 0 & 0 & 0 \\
2.86 & 8.49\times10^{-3} & -3.14\times10^{-4} & 0 & 0 & 0 & 0 \\
-1.11 & -61\times10^{-3} & 1.72\times10^{-4} & 0 & 0 & 0 & 0 \\
1.46\times10^{-1} & -26\times10^{-2} & 1.66\times10^{-6} & 0 & 0 & 0 & 0 \\
-2.22\times10^{-1} & 2.94\times10^{-2} & -3.24\times10^{-6} & 0 & 0 & 0 & 0 \\
-6.56\times10^{-2} & 2.75\times10^{-3} & 2.34\times10^{-5} & 0 & 0 & 0 & 0
\end{bmatrix} \tag{A.0.2}$$

$$\mathbf{M}_{50} = \begin{bmatrix}
1.16\times10^{-1} & 1.89\times10^{-3} & 37\times10^{-4} & 0 & 0 & 0 & 0 \\
27\times10^{-2} & -2.17\times10^{-4} & -2.15\times10^{-5} & -5.84\times10^{-4} & 3.55\times10^{-8} & -7.63\times10^{-6} & 2.63\times10^{-7} \\
1.31\times10^{-1} & 6.11\times10^{-2} & 0 & 0 & 0 & 0 & 0 \\
-1.8\times10^{-2} & 3.49\times10^{-4} & 2.88\times10^{-5} & 7.61\times10^{-4} & -3.14\times10^{-8} & -5.13\times10^{-6} & -2.17\times10^{-7} \\
-8.3\times10^{-2} & -1.68\times10^{-2} & 7.63\times10^{-5} & 0 & 0 & 0 & 0 \\
8.37\times10^{-3} & 1.53\times10^{-4} & 1.32\times10^{-5} & 8.65\times10^{-5} & -6.66\times10^{-9} & 2.2\times10^{-6} & -35\times10^{-7} \\
-5.37\times10^{-3} & 1.42\times10^{-4} & 6.1\times10^{-6} & 0 & 0 & 0 & 0 \\
8.5\times10^{-3} & -2.15\times10^{-4} & 31\times10^{-7} & 0 & 0 & 0 & 0 \\
1.41\times10^{-3} & 21\times10^{-4} & -6.89\times10^{-7} & 0 & 0 & 0 & 0 \\
-3.1\times10^{-2} & 0 & 0 & 0 & 0 & 0 & 0 \\
4.83\times10^{-1} & 6.49\times10^{-3} & -8.23\times10^{-5} & 0 & 0 & 0 & 0 \\
-6.88\times10^{-1} & -48\times10^{-3} & 17\times10^{-4} & 0 & 0 & 0 & 0 \\
-12\times10^{-1} & -2\times10^{-2} & 4.48\times10^{-5} & 0 & 0 & 0 & 0 \\
33\times10^{-2} & 1.76\times10^{-2} & 36\times10^{-5} & 0 & 0 & 0 & 0 \\
2.84 & 7.11\times10^{-3} & -38\times10^{-4} & 0 & 0 & 0 & 0 \\
-1.10 & -5.19\times10^{-3} & 1.69\times10^{-4} & 0 & 0 & 0 & 0 \\
1.24\times10^{-1} & -2\times10^{-2} & 4.82\times10^{-6} & 0 & 0 & 0 & 0 \\
-22\times10^{-1} & 2.83\times10^{-2} & 4.64\times10^{-6} & 0 & 0 & 0 & 0 \\
-73\times10^{-2} & 2.65\times10^{-3} & 1.24\times10^{-5} & 0 & 0 & 0 & 0
\end{bmatrix} \tag{A.0.3}$$

$$\mathbf{M}_{75} = \begin{bmatrix}
1.32\times10^{-1} & 23\times10^{-3} & 2.94\times10^{-4} & 0 & 0 & 0 & 0 \\
3.46\times10^{-2} & -3.86\times10^{-4} & -3.23\times10^{-5} & -5.84\times10^{-4} & 3.55\times10^{-8} & -7.63\times10^{-6} & 2.63\times10^{-7} \\
1.37\times10^{-1} & 63\times10^{-2} & 0 & 0 & 0 & 0 & 0 \\
-2.67\times10^{-2} & 6.14\times10^{-4} & 4.3\times10^{-5} & 7.61\times10^{-4} & -3.14\times10^{-8} & -5.13\times10^{-6} & -2.17\times10^{-7} \\
-9.36\times10^{-2} & -1.66\times10^{-2} & 8.62\times10^{-5} & 0 & 0 & 0 & 0 \\
1.18\times10^{-2} & 2.56\times10^{-4} & 1.81\times10^{-5} & 8.65\times10^{-5} & -6.66\times10^{-9} & 2.2\times10^{-6} & -35\times10^{-7} \\
-59\times10^{-3} & 1.38\times10^{-4} & 5.84\times10^{-6} & 0 & 0 & 0 & 0 \\
8.46\times10^{-3} & -1.86\times10^{-4} & -1.23\times10^{-6} & 0 & 0 & 0 & 0 \\
1.4\times10^{-3} & 25\times10^{-4} & -7\times10^{-7} & 0 & 0 & 0 & 0 \\
-2.76\times10^{-2} & 0 & 0 & 0 & 0 & 0 & 0 \\
4.82\times10^{-1} & 6.34\times10^{-3} & -8.16\times10^{-5} & 0 & 0 & 0 & 0 \\
-6.85\times10^{-1} & -3.68\times10^{-3} & 15\times10^{-4} & 0 & 0 & 0 & 0 \\
-11\times10^{-1} & -1.95\times10^{-2} & 3.95\times10^{-5} & 0 & 0 & 0 & 0 \\
4.62\times10^{-2} & 1.74\times10^{-2} & 9.14\times10^{-6} & 0 & 0 & 0 & 0 \\
2.82 & 5.74\times10^{-3} & -3\times10^{-4} & 0 & 0 & 0 & 0 \\
-19 & -4.51\times10^{-3} & 1.66\times10^{-4} & 0 & 0 & 0 & 0 \\
11\times10^{-1} & -1.96\times10^{-2} & 2.63\times10^{-5} & 0 & 0 & 0 & 0 \\
-1.82\times10^{-1} & 2.77\times10^{-2} & -8.53\times10^{-6} & 0 & 0 & 0 & 0 \\
-7.23\times10^{-2} & 2.67\times10^{-3} & -5.14\times10^{-5} & 0 & 0 & 0 & 0
\end{bmatrix} \tag{A.0.4}$$

$$\mathbf{M}_{100} = \begin{bmatrix}
1.51\times10^{-1} & 2.19\times10^{-3} & 2.77\times10^{-4} & 0 & 0 & 0 & 0 \\
5.1\times10^{-2} & -6.55\times10^{-4} & -4.28\times10^{-5} & -5.84\times10^{-4} & 3.55\times10^{-8} & -7.63\times10^{-6} & 2.63\times10^{-7} \\
1.45\times10^{-1} & 5.94\times10^{-2} & 0 & 0 & 0 & 0 & 0 \\
-3.67\times10^{-2} & 11\times10^{-3} & 5.61\times10^{-5} & 7.61\times10^{-4} & -3.14\times10^{-8} & -5.13\times10^{-6} & -2.17\times10^{-7} \\
-15\times10^{-1} & -1.63\times10^{-2} & 9.47\times10^{-5} & 0 & 0 & 0 & 0 \\
1.5\times10^{-2} & 41\times10^{-4} & 2.22\times10^{-5} & 8.65\times10^{-5} & -6.66\times10^{-9} & 2.2\times10^{-6} & -35\times10^{-7} \\
-4.74\times10^{-3} & 1.34\times10^{-4} & 5.58\times10^{-6} & 0 & 0 & 0 & 0 \\
8.43\times10^{-3} & -1.76\times10^{-4} & -1.2\times10^{-6} & 0 & 0 & 0 & 0 \\
1.39\times10^{-3} & 27\times10^{-4} & -6.71\times10^{-7} & 0 & 0 & 0 & 0 \\
-2.44\times10^{-2} & 0 & 0 & 0 & 0 & 0 & 0 \\
4.79\times10^{-1} & 6.25\times10^{-3} & -88\times10^{-5} & 0 & 0 & 0 & 0 \\
-6.8\times10^{-1} & -3.35\times10^{-3} & 14\times10^{-4} & 0 & 0 & 0 & 0 \\
-9.75\times10^{-2} & -1.93\times10^{-2} & 3.97\times10^{-5} & 0 & 0 & 0 & 0 \\
6.35\times10^{-2} & 1.71\times10^{-2} & -17\times10^{-5} & 0 & 0 & 0 & 0 \\
2.78 & 4.4\times10^{-3} & -2.88\times10^{-4} & 0 & 0 & 0 & 0 \\
-17 & -3.94\times10^{-3} & 1.63\times10^{-4} & 0 & 0 & 0 & 0 \\
7.36\times10^{-2} & -1.88\times10^{-2} & 1.29\times10^{-5} & 0 & 0 & 0 & 0 \\
-1.58\times10^{-1} & 2.69\times10^{-2} & -1.24\times10^{-5} & 0 & 0 & 0 & 0 \\
-7.85\times10^{-2} & 1.11\times10^{-3} & 6.19\times10^{-5} & 0 & 0 & 0 & 0
\end{bmatrix} \tag{A.0.5}$$

Payload coefficients are given in Table A.3. Coefficients not listed are assumed to be zero. The payload aerodynamic coefficients then reduce to the following, with angles in radians and moments referred to the payload mass centre.

$$C_L = C_{L\alpha}\alpha_L + C^2_{L\alpha}\alpha^2_L \tag{A.0.6}$$

$$C_m = C_{m\alpha}\alpha_L + C^2_{m\alpha}\alpha^2_L + C_{mq}\bar{q}_L \tag{A.0.7}$$

$$C_Y = C_{Y\beta}\beta_L \tag{A.0.8}$$

$$C_n = C_{n\beta}\beta_L \tag{A.0.9}$$

Coefficient	Value
$C_{Y\beta}$	-0.58
$C_{n\beta}$	0.06
$C_{L\alpha}$	1
$C_{m\alpha}$	0.25
C_{D0}	1
C_{DL}^2	0.06
C_{DY}^2	0.06
C_{mq}	-30

Table A.3 – Payload aerodynamic coefficients.

$$C_D = C_{D0} + C_{DL}C_L + C_{DL}^2 C_L^2 + C_{DY}^2 C_Y^2 \tag{A.0.10}$$

For a more in-depth discussion, the reader should consult [16].

Appendix B

Linear Model Details

B.1 System Matrices Computed With Numerical Linearisation

In Section 3.1.3, numerical linearisation was performed in order to compute the **F**, **G** and **W** matrices for the state space representation of the linear parafoil model. The computed matrices are as follows.

$$\mathbf{F} = \begin{bmatrix}
5.79\times10^{-2} & 2.39\times10^{-2} & -4.9\times10^{-3} & 2.21\times10^{-2} & -3.26 & 5.99\times10^{-2} & 6.94 & 3.16\times10^{-1} & -3.22\times10^{1} & 0 & 0 & 0 & 8.63 & -2.76\times10^{2} & 5.75 & -31\times10^{2} & 1.99\times10^{-6} & 0 \\
0 & -8.78\times10^{-1} & 0 & -6.19\times10^{-1} & 0 & 8.28\times10^{-2} & 0 & 9.94 & 0 & 0 & 0 & 0 & 3.17\times10^{2} & 0 & 8.14\times10^{1} & 0 & 3.12\times10^{2} & 0 \\
-7.9\times10^{-1} & 9.49\times10^{-3} & -1.15\times10^{-1} & 8.76\times10^{-3} & 1.23\times10^{1} & 2.38\times10^{-2} & -1.99\times10^{1} & 1.25\times10^{-1} & 7.83\times10^{1} & 0 & 0 & 0 & 3.42 & 7.16\times10^{1} & 2.28 & 1.68\times10^{2} & -1.71\times10^{-6} & 0 \\
0 & 1.42 & 0 & 3.69\times10^{-1} & 0 & -32 & 0 & -8.91 & 0 & 0 & 0 & 0 & -5.11\times10^{2} & 0 & 1.94\times10^{2} & 0 & -7.23\times10^{1} & 0 \\
-5.47\times10^{-1} & -1.25\times10^{-1} & -1.16 & -1.15\times10^{-1} & 6 & -3.12\times10^{-1} & -1.9\times10^{1} & -1.65 & 11\times10^{2} & 0 & 0 & 0 & -4.49\times10^{1} & -2.62\times10^{2} & -2.99\times10^{1} & 5.24\times10^{-1} & -7.65\times10^{-5} & 2.29\times10^{-1} \\
0 & -2.19 & 0 & -8.27\times10^{1} & 0 & 3.59 & 0 & 11\times10^{2} & 0 & 0 & 0 & 0 & 7.88\times10^{2} & 0 & -1.24\times10^{4} & 0 & 8.69\times10^{2} & 0 \\
-4.34\times10^{-1} & -12\times10^{-1} & -1.16 & -9.43\times10^{-2} & 95 & -2.56\times10^{-1} & -24\times10^{1} & -1.35 & 7.89\times10^{1} & 0 & 0 & 0 & -3.68\times10^{1} & -2.9\times10^{2} & -2.46\times10^{1} & -2.2\times10^{2} & -7.42\times10^{-5} & 2.29\times10^{-1} \\
0 & 4.71\times10^{-1} & 0 & 41\times10^{-1} & 0 & -4.11\times10^{-1} & 0 & -13\times10^{1} & 0 & 0 & 0 & 0 & -1.7\times10^{2} & 0 & 9.73\times10^{1} & 0 & -1.32\times10^{2} & 0 \\
3.16\times10^{-1} & -8.13\times10^{-4} & -1.59\times10^{1} & -1.18\times10^{-3} & 1.93\times10^{1} & -1.64\times10^{-3} & -2.38\times10^{1} & -9.27\times10^{-3} & -26\times10^{2} & 0 & 0 & 0 & -2.39\times10^{-1} & -5.78\times10^{3} & -2.59\times10^{-1} & -5.62\times10^{3} & -4.29\times10^{-6} & -5.1\times10^{4} \\
3.9\times10^{-2} & 0 & -1.19\times10^{-6} & 0 & 6.4\times10^{-4} & 0 & 8.47\times10^{-3} & 0 & -1.26\times10^{-3} & 0 & 0 & 0 & 5.33\times10^{-8} & -4.42\times10^{-1} & -3.55\times10^{-8} & -4.37\times10^{-1} & -3.55\times10^{-8} & 0 \\
0 & 3.88\times10^{-2} & 0 & 1.43\times10^{-4} & 0 & -7.19\times10^{-3} & 0 & 3.98\times10^{-4} & 0 & 0 & 0 & 0 & -15 & 0 & 3.24\times10^{-1} & 0 & 3.56\times10^{-8} & 0 \\
-1.2\times10^{-6} & 0 & 3.91\times10^{-2} & 0 & 21\times10^{-5} & 0 & 2.65\times10^{-4} & 0 & 3.9\times10^{-2} & 0 & 0 & 0 & 0 & 1.13 & 8.88\times10^{-9} & 1.13 & 0 & 0 \\
0 & 1.43\times10^{-4} & 0 & 5.48\times10^{-1} & 0 & -2.39\times10^{-2} & 0 & -5.46\times10^{-1} & 0 & 0 & 0 & 0 & -5.17\times10^{-2} & 0 & 8.12\times10^{1} & 0 & 3.28\times10^{-8} & 0 \\
6.4\times10^{-4} & 0 & 2\times10^{-5} & 0 & 79\times10^{-1} & 0 & -8.61\times10^{-1} & 0 & -4.12\times10^{-2} & 0 & 0 & 0 & 28\times10^{-7} & -8.43\times10^{-2} & -2.63\times10^{-6} & 5.47 & 19\times10^{-7} & 0 \\
0 & -7.19\times10^{-3} & 0 & -2.39\times10^{-2} & 0 & 1.88\times10^{-2} & 0 & 2.24\times10^{-2} & 0 & 0 & 0 & 0 & 2.59 & 0 & -4.28 & 0 & -8.91\times10^{-8} & 0 \\
8.47\times10^{-3} & 0 & 2.65\times10^{-4} & 0 & -8.61\times10^{-1} & 0 & 17 & 0 & 51\times10^{-2} & 0 & 0 & 0 & -6.49\times10^{-7} & -1.12 & -6.64\times10^{-6} & -7.86 & -2.43\times10^{-7} & 0 \\
0 & 3.98\times10^{-4} & 0 & -5.46\times10^{-1} & 0 & 2.24\times10^{-2} & 0 & 8.5\times10^{-1} & 0 & 0 & 0 & 0 & -1.43\times10^{-1} & 0 & -88\times10^{1} & 0 & -2.61\times10^{-8} & 0 \\
-1.26\times10^{-3} & 0 & 3.9\times10^{-2} & 0 & -4.12\times10^{-2} & 0 & 51\times10^{-2} & 0 & 4.9\times10^{-1} & 0 & 0 & 0 & 2.66\times10^{-8} & 1.43\times10^{1} & -4.88\times10^{-7} & 1.39\times10^{1} & 2.66\times10^{-8} & 0
\end{bmatrix}$$

$$\mathbf{G} = \begin{bmatrix}
-2.77\times10^{-1} & -2.77\times10^{-1} \\
-2.42\times10^{-1} & 2.42\times10^{-1} \\
-2.9\times10^{-1} & -2.9\times10^{-1} \\
-3.22\times10^{-1} & 3.22\times10^{-1} \\
1.13 & 1.13 \\
-6.84\times10^{-1} & 6.84\times10^{-1} \\
8.6\times10^{-1} & 8.6\times10^{-1} \\
-2.87\times10^{-1} & 2.87\times10^{-1} \\
3\times10^{-2} & 3\times10^{-2} \\
0 & 0 \\
0 & 0 \\
0 & 0 \\
0 & 0 \\
0 & 0 \\
0 & 0 \\
0 & 0 \\
0 & 0 \\
0 & 0 \\
0 & 0
\end{bmatrix}$$

$$\mathbf{W} = \begin{bmatrix}
2.82\times10^{1} & 1.78\times10^{-7} & -7.12\times10^{1} \\
0 & 2.38\times10^{1} & 0 \\
-2.95\times10^{1} & 0 & 1.78\times10^{2} \\
0 & -5.54 & 0 \\
-1\times10^{2} & -9.95\times10^{-7} & 2.56\times10^{2} \\
0 & 63\times10^{1} & 0 \\
-84\times10^{1} & -8.53\times10^{-7} & 26\times10^{2} \\
0 & -8.66 & 0 \\
2.22 & 3.87\times10^{-9} & -1.39\times10^{1} \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0
\end{bmatrix}$$

References

[1] Ward, M.: *Adaptive Glide Slope Control for Parafoil and Payload Aircraft.* Ph.D. thesis, School of Aerospace Engineering, Georgia Institute of Technology, May 2012.

[2] Slegers, N., Beyer, E. and Costelllo, M.: Use of dynamic incidence angle for glide slope control of autonomous parafoils. In: *19th AIAA Aerodynamic Decelerator Systems Technology Conference and Seminar.* May 2007.

[3] Slegers, N. and Costello, M.: Aspects of control for a parafoil and payload system. *Journal of Guidance, Control, and Dynamics*, vol. 26, no. 6, pp. 898–905, 2003.

[4] Gavrilovski, A., Ward, M. and Costello, M.: Parafoil glide slope control using canopy spoilers. American Institute of Aeronautics and Astronautics, 2011.

[5] Slegers, N.J.: Effects of canopy-payload relative motion on control of autonomous parafoils. *Journal of guidance, control, and dynamics*, vol. 33, no. 1, pp. 116–125, 2010.

[6] Strickert, G. and Jann, T.: Determination of the relative motion between parafoil canopy and load using advanced video image processing techniques. *AIAA Paper*, vol. 1754, pp. 8–11, 1999.

[7] Lingard, J.: Ram-air parachute design. In: *Precision Aerial Delivery Seminar, 13th AIAA Aerodynamic Decelerator Systems Technology Conference, Clearwater Beach.* 1995.

[8] Yakimenko, O.A. and Slegers, N.J.: Optimal control for terminal guidance of autonomous parafoils. In: *20th AIAA Aerodynamic Decelerator Systems Technology conference and Seminar.* May 2009.

[9] Toohey, D.: *Development of a Small Parafoil Vehicle for Precision Delivery.* Master's thesis, Massachusetts Institute of Technology, 2005.

[10] Jann, T.: Advanced features for autonomous parafoil guidance, navigation and control. In: *18th AIAA Aerodynamic Decelerator Systems Technology Conference and Seminar.* 2005.

[11] Rademacher, B.J.: *In-flight trajectory planning and guidance for autonomous parafoils.* Ph.D. thesis, Iowa State University, 2009.

[12] Roos, J.-C.: *Autonomous Take-Off and Landing of a Fixed Wing Unmanned Aerial Vehicle.* Master's thesis, Stellenbosch University, 2007.

[13] Ward, M., Costello, M. and Slegers, N.: Specialized system identification for parafoil and payload systems. *Journal of Guidance, Control, and Dynamics*, vol. 35, no. 2, pp. 588–597, 2012.

[14] Carter, D.W., George, S., Hattis, P.D., McConley, M.W., Rasmussen, S.A., Singh, L. and Tavan, S.: Autonomous large parafoil guidance, navigation and control system design status. In: *19th AIAA Aerodynamic Decelerator Systems Technology Conference Seminar*. May 2007.

[15] Yakimenko, O.A., Slegers, N.J. and Taiaden, R.A.: Development and testing of the miniature aerial delivery system snowflake. In: *20th AIAA Aerodynamic Decelerator Systems Technology Conference and Seminar*. May 2009.

[16] Redelinghuys, C.: A flight simulation algorithm for a parafoil suspending an air vehicle. *Journal of guidance, control, and dynamics*, vol. 30, no. 3, pp. 791–803, 2007.

[17] Grunwald, A.J., Redelinghuys, C., Rhodes, S., Norton, W., Adams, J., Booysen, T. and Pead, J.: Reconfigurable flight testing system for parafoils. 1 July 2011. A chronological account of the design considerations and phases of development of a remotely controlled parafoil system.

[18] Peddle, I.: *Autonomous Flight of a Model Aircraft*. Master's thesis, Stellenbosch University, 2005.

[19] Franklin, G.F., Powell, J.D. and Workman, M.L.: *Digital Control of Dynamic Systems*. 3rd edn. Addison-Wesley, 1998.

[20] Persson, J., Slootweg, J., Rouco, L., Söder, L. and Kling, W.: A comparison of eigenvalues obtained with two dynamic simulation software packages. In: *2003 IEEE Bologna Power Tech Conference*. June 2003.

[21] Bryson, A.E.: *Control of Spacecraft and Aircraft*. Princeton University Press, 1994.

[22] Redelinghuys, C.: A study of the linearized dynamics of a parafoil and UAV configuration. 2006. Summary of a published study by S. Müller, 2002.

[23] Cook, M.V.: *Flight Dynamics Principles*. Butterworth-Heinermann, 1997.

[24] Hur, G.-B.: *Identification Of Powered Parafoil-Vehicle Dynamics From Modelling And Flight Test Data*. Ph.D. thesis, Texas A&M University, May 2005.

[25] Park, S.: *Avionics and Control System Development for Mid-Air Rendezvous of Two Unmanned Aerial Vehicles*. Ph.D. thesis, Massachusetts Institute of Technology, 2004.

[26] Yakimenko, O.A.: Direct method for rapid prototyping of near-optimal aircraft trajectories. *Journal of Guidance, Control, and Dynamics*, vol. 23, no. 5, pp. 865–875, 2000.

[27] Blaauw, D.: *Flight Control System for a Variable Stability Blended-Wing Unmanned Aerial Vehicle*. Master's thesis, Stellenbosch University, 2009.

[28] Venter, J.: *Development of an Experimental Tilt-wing VTOL Unmanned Aerial Vehicle*. Master's thesis, Stellenbosch University, 2005.

[29] Hough, W.: *Autonomous aerobatic flight of a fixed wing unmanned aerial vehicle*. Master's thesis, Stellenbosch University, December 2008.

[30] de Jager, A.M.: *The design and implementation of vision-based autonomous rotorcraft landing*. Master's thesis, Stellenbosch University, March 2011.

[31] Silberbauer, M.: Simulation visualisation system. Undergraduate final year project, Stellenbosch University, 2005.